Adaptation to data shift without labels

Methods, benchmarks and domain adaptation at test time with optimal transport on biomedical signals.

Rémi Flamary - CMAP, École Polytechnique, Institut Polytechnique de Paris

July 1rst 2025

CAp 2025, PFIA 2025, Dijon.

Sélection de variables pour l'apprentissage simultanée de tâches

R. Flamary, A. Rakotomamonjy , G. Gasso, S. Canu

LITIS EA 4108, INSA-Université de Rouen
76800 Saint Etienne du Rouvray, France

Résumé : Cette article traite de la sélection de variables pour l'apprentissage simultanée de taches de discrimination SVM . Nous formulons ce problème comme étant un apprentissage multi-taches avec pour terme de régularisation une norme mixte de type $\ell_p - \ell_2$ avec $p \leq 1$. Cette dernière permet d'obtenir des modèles de discrimination pour chaque tâche, utilisant un même sous-ensemble des variables. Nous proposons tout d'abord un algorithme permettant de résoudre le problème d'apprentissage lorsque la norme mixte est convexe ($p = 1$). Ensuite, à l'aide de la programmation DC, nous traitons le cas non-convexe ($p < 1$) . Nous montrons que ce dernier cas peut être résolu par un algorithme itératif où, à chaque itération, un problème basé sur la norme mixte $\ell_1 - \ell_2$ est résolu. Nos expériences montrent l'interêt de la méthode sur quelques problèmes de discriminations simultanées.

Mots-clés : Apprentissage multi-tâches, Sélection de variables, méthodes à noyaux

Sélection de variables pour l'apprentissage simultanée de tâches

R. Flamary, A. Rakotomamonjy , G. Gasso, S. Canu

LITIS EA 4108, INSA-Université de Rouen
76800 Saint Etienne du Rouvray, France

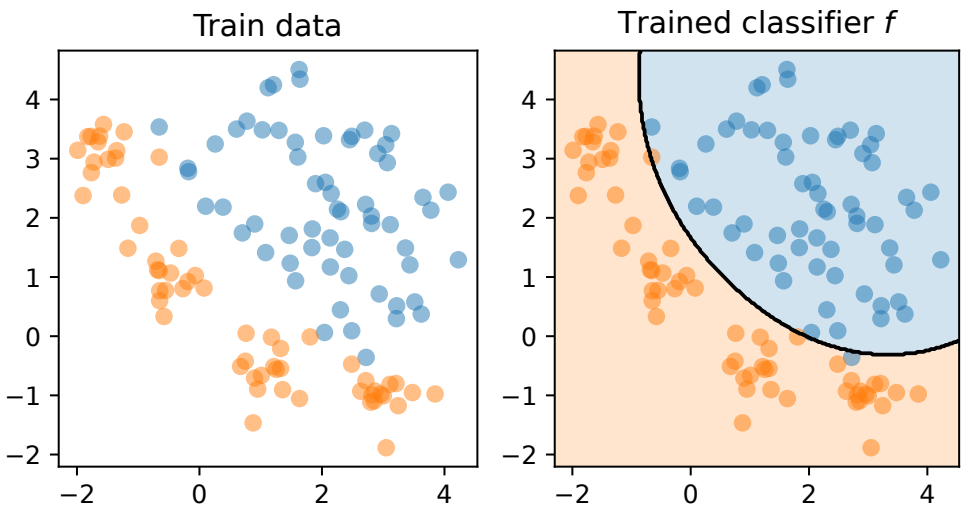
Résumé : Cette article traite de la sélection de variables pour l'apprentissage simultanée de taches de discrimination SVM . Nous formulons ce problème comme étant un apprentissage multi-taches avec pour terme de régularisation une norme mixte de type $\ell_p - \ell_2$ avec $p \leq 1$. Cette dernière permet d'obtenir des modèles de discrimination pour chaque tâche, utilisant un même sous-ensemble des variables. Nous proposons tout d'abord un algorithme permettant de résoudre le problème d'apprentissage lorsque la norme mixte est convexe ($p = 1$). Ensuite, à l'aide de la programmation DC, nous traitons le cas non-convexe ($p < 1$) . Nous montrons que ce dernier cas peut être résolu par un algorithme itératif où, à chaque itération, un problème basé sur la norme mixte $\ell_1 - \ell_2$ est résolu. Nos expériences montrent l'interêt de la méthode sur quelques problèmes de discriminations simultanées.

Mots-clés : Apprentissage multi-tâches, Sélection de variables, méthodes à noyaux

Table of content

Domain Adaptation or adapting to data shift without labels

Supervised learning



Traditional supervised learning

$$\min_{f} \quad \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\widehat{\mathcal{P}}} \mathcal{L}(y, f(\mathbf{x})) = \frac{1}{n}\sum_{j} \mathcal{L}(y_j, f(\mathbf{x}_j)) \tag{1}$$

- We want to learn predictor such that $y \approx f(\mathbf{x})$.
- Training dataset $(\mathbf{x}_i, y_i)_{i=1,\dots,n} \sim \mathcal{P}(X,Y)^n$ (Training dist. $\widehat{\mathcal{P}}(X,Y)$).
- The loss function $\mathcal{L}(y, f(\mathbf{x}))$ measures the prediction error.
- Generalization to new data when the training and test distributions are the same.
- Regularization (explicit or implicit) can be used to avoid overfitting.

shift happens!

Data shift

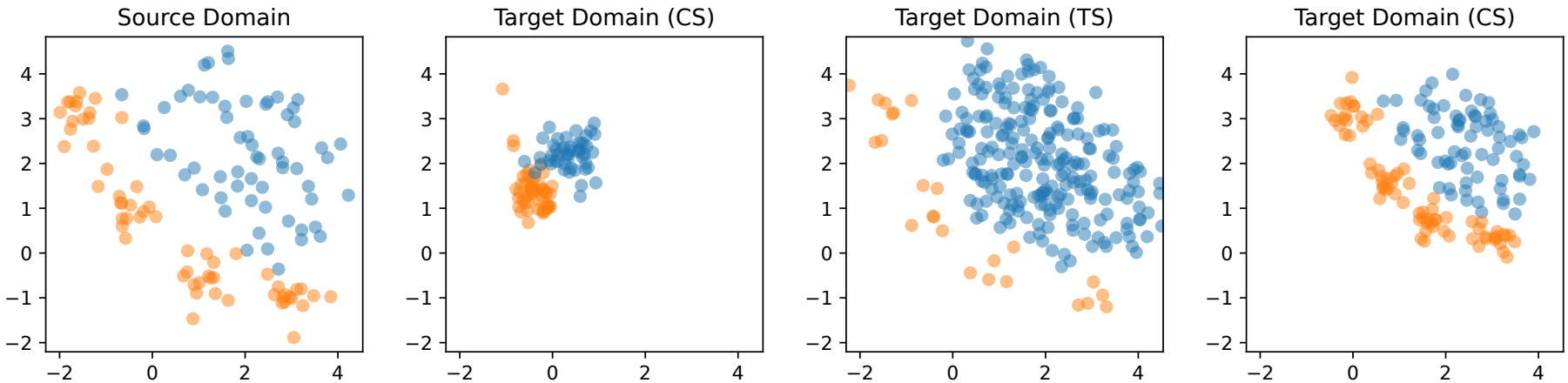
Data shift examples : $\mathcal{P}^s \neq \mathcal{P}^t$

- A classifier learned on $\mathcal{P}^s$ (Source) might fail on $\mathcal{P}^t$ (Target).
- Data shift examples :
  - Change of the sensor or acquisition protocol.
  - Change of the environment (e.g. weather, light, ...).
  - Simulation vs real data.

Domain Adaptation (DA)

- Aim at learning a function f that works on $\mathcal{P}^t$ using data samples from $\mathcal{P}^s$.
- Unsupervised DA : also use target samples $\mathbf{x}^t$ from $\mathcal{P}^t$ but with no labels.
- Ill-posed problem without any assumption on the shift.

Families of data shift



Domain Adaptation strategy

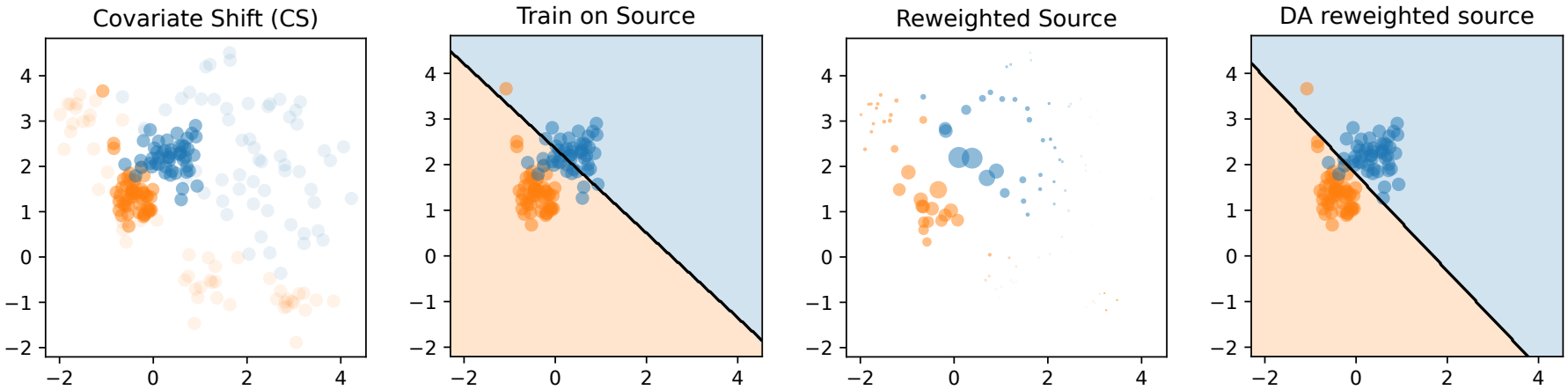
1. Model the shift between $\mathcal{P}^s$ and $\mathcal{P}^t$ and estimate it.
2. Train on the labeled adapted (shifted) source data.

Alternative: train a predictor that is invariant to the shift.

Most common data shifts [Moreno-Torres et al., 2012]

- **Covariate shift**: $P^s_{\mathcal{X}}(\mathbf{x}) \neq P^t_{\mathcal{X}}(\mathbf{x})$, $P^s(y|\mathbf{x}) = P^t(y|\mathbf{x})$
- **Target shift**: $P^s_{\mathcal{Y}}(y) \neq P^t_{\mathcal{Y}}(y)$, $P^s(\mathbf{x}|y) = P^t(\mathbf{x}|y)$
- **Conditional shift**: $P^s(y|\mathbf{x}) \neq P^t(y|\mathbf{x})$ or $P^s(\mathbf{x}|y) \neq P^t(\mathbf{x}|y)$
- **Domain Invariant Sub.**: $P^s(y|\mathbf{W}\mathbf{x}) = P^t(y|\mathbf{W}\mathbf{x})$ or $P^s(y|g(\mathbf{x})) = P^t(y|g(\mathbf{x}))$

Covariate Shift (CS)



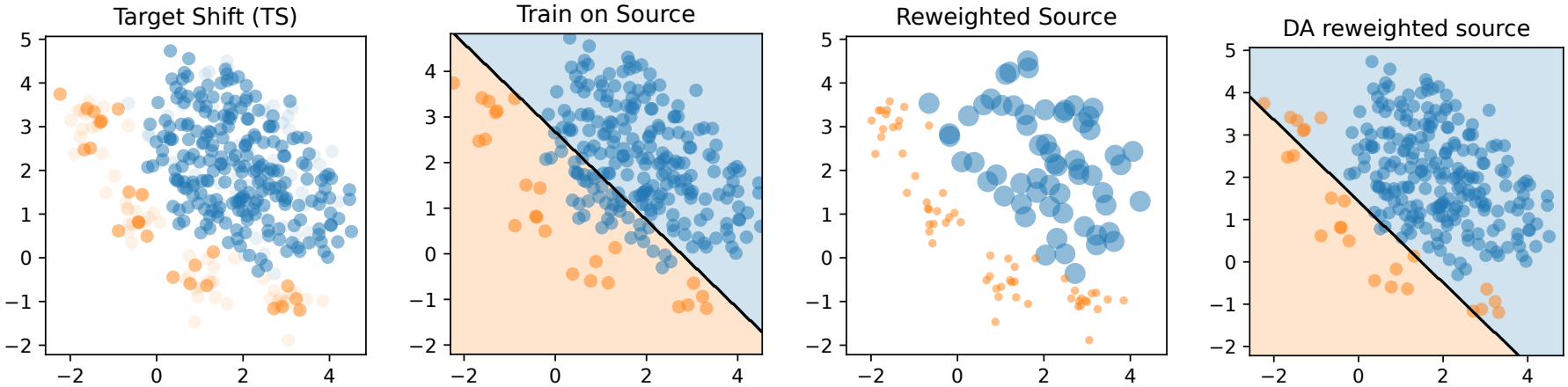
Principle and methods

- **Assumption** : $P^s_{\mathcal{X}}(\mathbf{x}) \neq P^t_{\mathcal{X}}(\mathbf{x})$, $P^s(y|\mathbf{x}) = P^t(y|\mathbf{x})$
- Can be exactly compensated by training on reweighted source samples with w

$$w(\mathbf{x}) = \frac{P^t_{\mathcal{X}}(\mathbf{x})}{P^s_{\mathcal{X}}(\mathbf{x})} \tag{2}$$

- **Existing approaches for estimating** $\hat{w}$:
  - Gaussian Approximation [Shimodaira, 2000] or kernel density estimation [Sugiyama et al., 2005].
  - Divergence minimization with MMD for Kernel Mean Matching (KMM) [Huang et al., 2006, Gretton et al., 2009] or KL (KLIEP) [Sugiyama et al., 2007].
  - Use Source/target classifier $\hat{w}(\mathbf{x}) \propto P(\text{domain} = \text{target}|\mathbf{x})$ [Sugiyama et al., 2012].

Target Shift (TS)



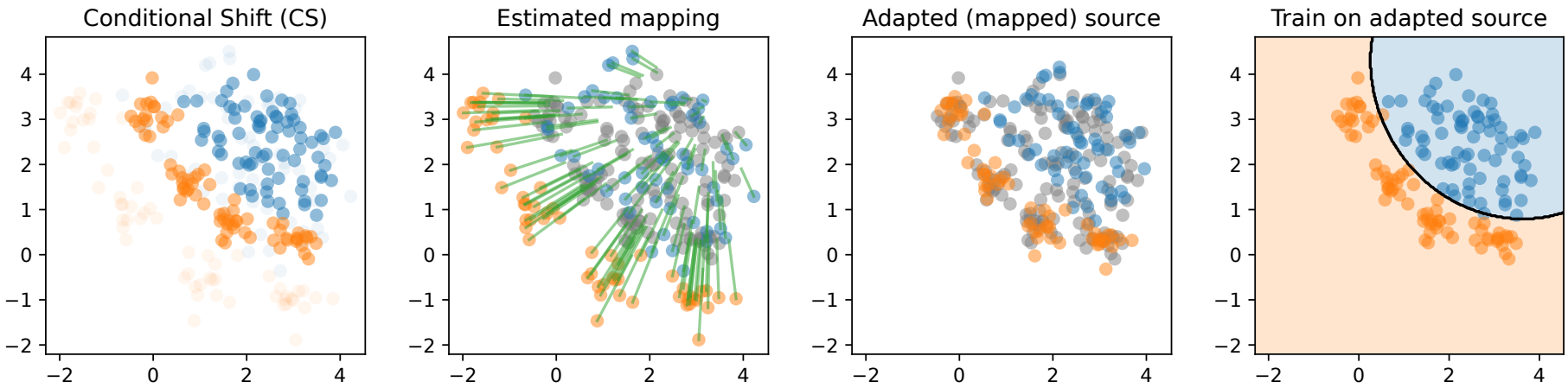
Principle and methods (a.k.a prior shift or label shift)

- **Assumption** : $P^s_{\mathcal{Y}}(y) \neq P^t_{\mathcal{Y}}(y)$, $P^s(\mathbf{x}|y) = P^t(\mathbf{x}|y)$
- Can be compensated by training on class-reweighted source samples with w

$$w(y) = \frac{P^t_{\mathcal{Y}}(y)}{P^s_{\mathcal{Y}}(y)} \tag{3}$$

- **Existing approaches for estimating** $\hat{w}$:
  - Black Box Shift Estimation (BBSE) [Lipton et al., 2018] uses a pre-trained trained classifier h and its confusion matrix on source to estimate $w(y)$.
  - $\widehat{\mathcal{P}}^t_{\mathcal{Y}}(y)$ can be estimated by divergence minimization such as Kernel Mean Matching [Zhang et al., 2013] or Wasserstein distance [Redko et al., 2019].

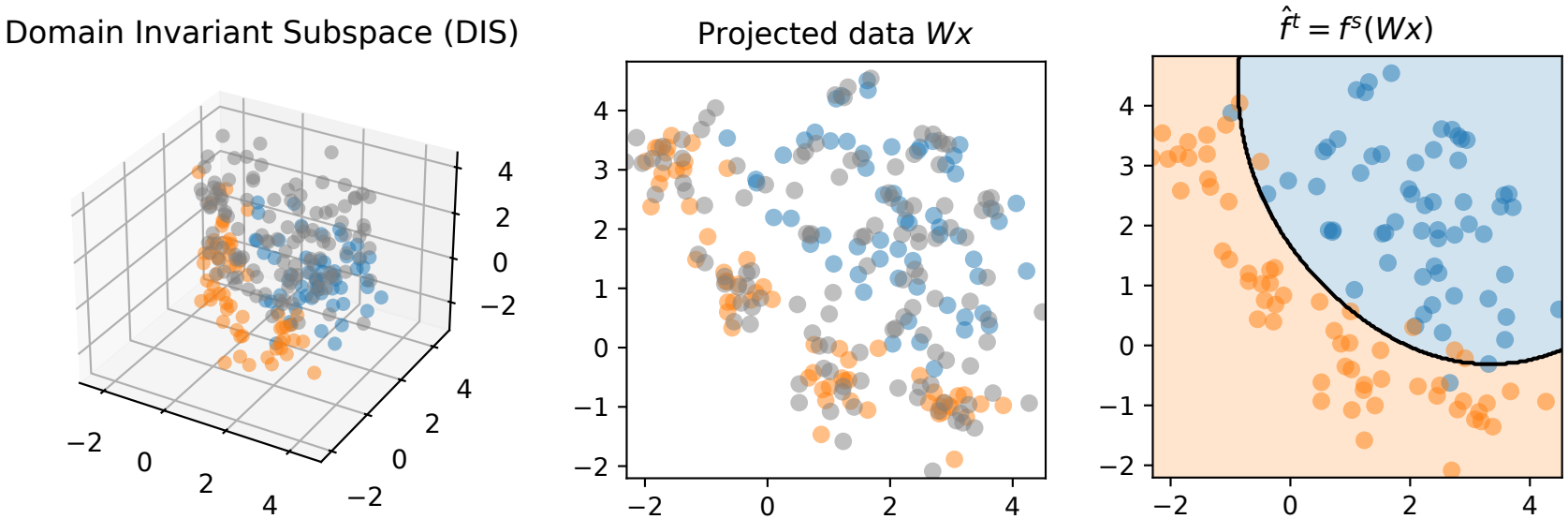
Conditional Shift



Principle and methods (a.k.a concept drift)

- **Assumption** : $P^s(y|\mathbf{x}) \neq P^t(y|\mathbf{x})$ or $P^s(\mathbf{x}|y) \neq P^t(\mathbf{x}|y)$
- **Strategy** for model $P^s(y|m(\mathbf{x})) = P^t(y|\mathbf{x})$:
  1. Model the shift m and estimate it from the data.
  2. Train on adapted source samples $\{m(\mathbf{x}_i^s), y_i^s\}_i$
- **Existing approaches for estimating** m:
  - Correlation alignment (CORAL) [Sun and Saenko, 2016] : $m(\mathbf{x}) = \Sigma_t^{1/2}\Sigma_s^{-1/2}\mathbf{x}$
  - Optimal Transport (OTDA) [Courty et al., 2016] : $m(\mathbf{x})$ is the regularized OT mapping between $\mathcal{P}_{\mathcal{X}}^s$ and $\mathcal{P}_{\mathcal{X}}^t$.
  - Linear OT mapping [Flamary et al., 2019] : $m(\mathbf{x}) = A\mathbf{x} + b$
  - Estimate affine mapping by minimizing $\text{MMD}(\mathcal{P}_{\mathcal{X}}^t, m\#\mathcal{P}_{\mathcal{X}}^s)$ [Zhang et al., 2013].

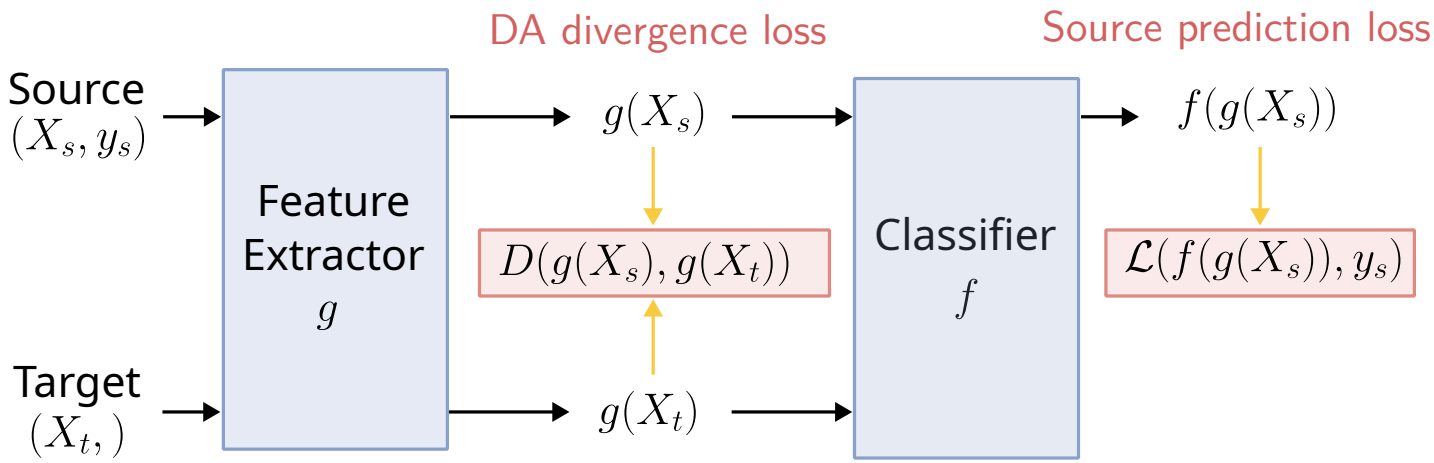
Domain Invariant Subspaces (DIS)



Principle and methods

- **Assumption**: $\exists \mathbf{W}$ such that $P^s(y|\mathbf{Wx}) = P^t(y|\mathbf{Wx})$
- **Strategy**:
  1. Estimate the projection $\mathbf{W}$ operator.
  2. Train predictor f_p on the projected source samples $\{\mathbf{Wx}_i^s, y_i^s\}_i$.
  3. Predict on target with $\hat{f}^t(\mathbf{x}) = f_p(\mathbf{Wx})$.
- **Existing approaches for estimating the subspace**:
  - Subspace alignment (SA) [Fernando et al., 2013] performs PCA in each domains.
  - Transfer Component Analysis (TCA) [Pan et al., 2010] uses MMD to estimate $\mathbf{W}$.
  - Transfer Subspace Learning (TSL) [Si et al., 2010] uses a KL divergence between the source and target distributions.

Deep DA : Domain invariant feature learning



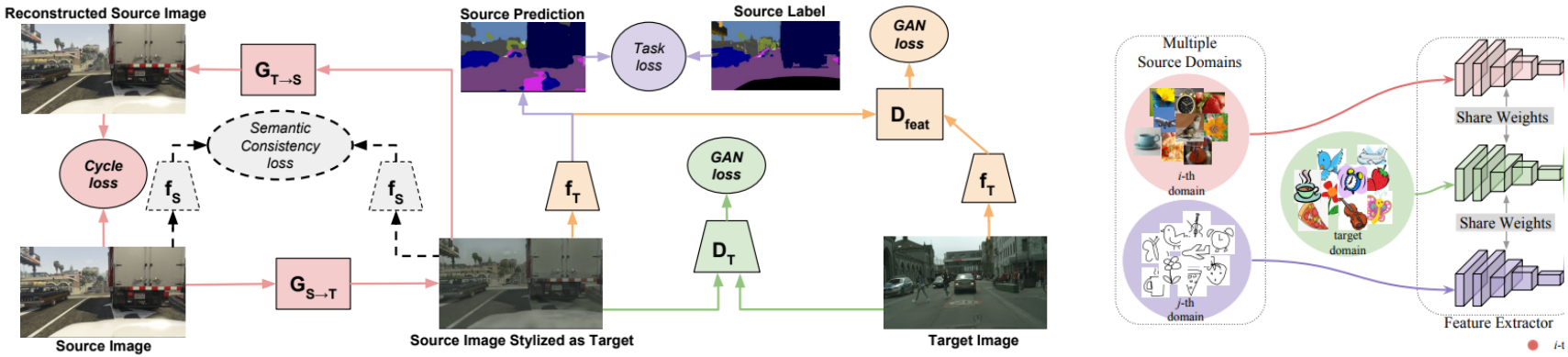
Principle and methods

- **Assumption**: $\exists g$ such that such that $P^s(y|g(\mathbf{x})) = P^t(y|g(\mathbf{x}))$.
- **Classical strategy for deep DA**:

$$\min_{f,g} \underbrace{\frac{1}{n_s}\sum_{i=1}^{n_s} L(y_i^s, f(g(\mathbf{x}_i^s)))}_{\text{Loss on source}} + \lambda \underbrace{D(g\#\hat{\mathcal{P}}_{\mathcal{X}}^s, g\#\hat{\mathcal{P}}_{\mathcal{X}}^t)}_{\text{Disc. on feature marginals}} \tag{4}$$

- **Existing approaches**:
  - Domain Adversarial Neural Networks (DANN) [Ganin et al., 2016].
  - Deep CORAL [Sun and Saenko, 2016] minimizes difference between covariances.
  - Deep Domain Confusion (DDC) [Tzeng et al., 2014] uses MMD.
  - DeepJDOT [Damodaran et al., 2018] OT loss on joint dist.

Other Deep DA strategies



Other strategies

- CycleGAN [Zhu et al., 2017] learns a mapping between domains with GANs.
- Constrastive Adaptation Networks (CAN) [Kang et al., 2019].

Using multiple domains for invariant representations

- Moment Matching for Multi-source DA (M^3SDA) [Peng et al., 2019] estimates invariant representation and then perform weighting of source classifier.
- Wasserstein Barycenter Transport (WBT) [Montesuma and Mboula, 2021] computes Wasserstein barycenter of source domains and then performs OTDA.
- Domain Generalization (DG) aim at learning a model that generalizes to unseen domains (Survey in [Zhou et al., 2022]).

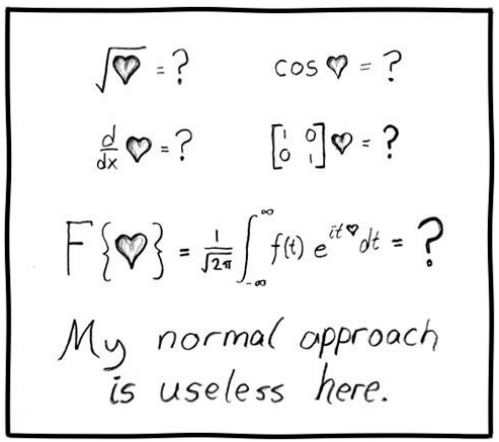
SKADA : SciKit ADAptation

SKADA : Scikit ADAptation : `https://scikit-adaptation.github.io/`

- A Python library for Domain Adaptation.
- Implements many DA methods (~ 20 shallow, ~ 10 deep).
- Easy to use, with a scikit-learn and Pytorch compatible API.
- Available on GitHub: `https://github.com/scikit-adaptation/skada/`

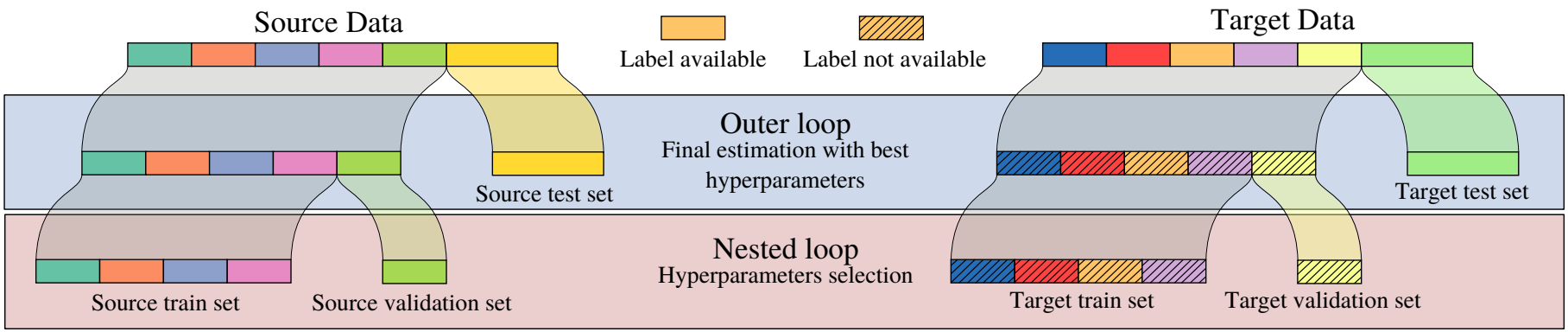
Why is DA not used all the time ?

- Type of shift not always known.
- Many methods, many parameters.
- How to choose?
- No target labels available for validation.



Realistic DA benchmark : SKADA-bench

Realistic DA benchmark : SKADA-bench



Benchmark objectives [Lalou et al., 2025]

- Compare DA methods on realistic data shifts from multiple modalities.
- Provide a realistic validation procedure and performance estimation:
  - Use nested cross-validation to select the best parameters.
  - No target labels available during validation.
  - Only use DA validation scorers that do not use target labels.
- Fully reproducible, using Benchopt [Moreau et al., 2022].
- Open source and easy to extend and run for new datasets and methods:

  `https://github.com/scikit-adaptation/skada-bench`

Benchmark datasets

Dataset	Modality	Preprocessing	# adapt	# classes	# samples	# features
`Office 31`	CV	Decaff + PCA	6	31	470 ± 350	100
`Office Home`	CV	ResNet + PCA	12	65	3897 ± 850	100
`MNIST/USPS`	CV	Vect + PCA	2	10	3000 / 10000	50
`20 Newsgroup`	NLP	LLM + PCA	6	2	3728 ± 174	50
`Amazon Review`	NLP	LLM + PCA	12	4	2000	50
`Mushrooms`	Tabular	One Hot Encoding	2	2	4062 ± 546	117
`Phishing`	Tabular	NA	2	2	5527 ± 1734	30
`BCI`	Biosignals	Cov+TS	9	4	288	253

Datasets

- Different modalities (Images, text, tabular, signals).
- Pre-processing done using modern feature extraction methods depending on the modality (PCA, pre-trained models, LLMs).
- We use as adaptation tasks all pairs of domains in each dataset.
- Models can be trained on a domain with good accuracy.
- There is a shift betwee the domains (see next slide).

Benchmark results : the big table

		Cov. shift	Tar. shift	Cond. shift	Sub. shift	Office31	OfficeHome	MNIST/USPS	20NewsGroups	AmazonReview	Mushrooms	Phishing	BCI	Selected Scorer	Rank
	Train Src	0.88	0.85	0.66	0.19	0.65	0.56	0.54	0.59	0.7	0.72	0.91	0.55		10.66
	Train Tgt	0.92	0.93	0.82	0.98	0.89	0.8	0.96	1.0	0.73	1.0	0.97	0.64		1.55
Reweighting	Dens. RW	0.88	0.86	0.66	0.18	0.62	0.56	0.54	0.58	0.7	0.71	0.91	0.55	IW	12.20
	Disc. RW	0.85	0.83	0.71	0.18	0.63	0.54	0.5	0.6	0.68	0.75	0.91	0.56	CircV	8.75
	Gauss. RW	0.89	0.86	0.65	0.21	0.22	0.44	0.11	0.54	0.55	0.51	0.46	0.25	CircV	16.45
	KLIEP	0.88	0.86	0.66	0.19	0.65	0.56	0.54	0.6	0.69	0.72	0.91	0.55	CircV	10.56
	KMM	0.89	0.85	0.64	0.16	0.64	0.54	0.52	0.7	0.57	0.74	0.91	0.52	CircV	11.74
	NN RW	0.89	0.86	0.67	0.15	0.65	0.55	0.54	0.59	0.66	0.71	0.91	0.54	CircV	9.15
	MMDTarS	0.88	0.86	0.64	0.2	0.6	0.56	0.54	0.59	0.7	0.74	0.91	0.55	IW	10.81
Mapping	CORAL	0.74	0.7	0.76	0.18	0.65	0.57	0.62	0.73	0.7	0.72	0.92	0.62	CircV	5.08
	MapOT	0.72	0.57	0.82	0.02	0.6	0.51	0.61	0.76	0.68	0.63	0.84	0.47	PE	10.21
	EntOT	0.71	0.6	0.82	0.12	0.64	0.58	0.6	0.83	0.62	0.75	0.86	0.54	CircV	9.40
	ClassRegOT	0.74	0.58	0.81	0.11	NA	0.53	0.62	0.97	0.68	0.82	0.89	0.52	IW	8.25
	LinOT	0.73	0.73	0.76	0.18	0.66	0.57	0.64	0.82	0.7	0.76	0.91	0.61	CircV	4.06
	MMD-LS	0.78	0.72	0.76	0.56	0.65	0.56	0.55	0.97	0.63	0.85	NA	0.5	MixVal	8.22
Subspace	JPCA	0.88	0.85	0.66	0.15	0.62	0.48	0.51	0.77	0.69	0.78	0.9	0.54	PE	8.98
	SA	0.74	0.68	0.8	0.11	0.65	0.57	0.56	0.88	0.67	0.78	0.89	0.53	CircV	7.80
	TCA	0.52	0.47	0.51	0.62	0.04	0.02	0.07	0.61	0.61	0.49	0.48	0.26	DEV	17.58
	TSL	0.88	0.85	0.66	0.2	0.63	0.48	0.45	0.63	0.69	0.45	0.89	0.26	IW	15.09
Other	JDOT	0.72	0.58	0.82	0.13	0.6	0.42	0.59	0.79	0.67	0.65	0.79	0.47	IW	11.42
	OTLabelProp	0.72	0.59	0.8	0.07	0.66	0.56	0.62	0.86	0.67	0.64	0.86	0.5	CircV	10.01
	DASVM	0.89	0.86	0.65	0.15	NA	NA	NA	0.87	NA	0.83	0.85	NA	MixVal	7.29

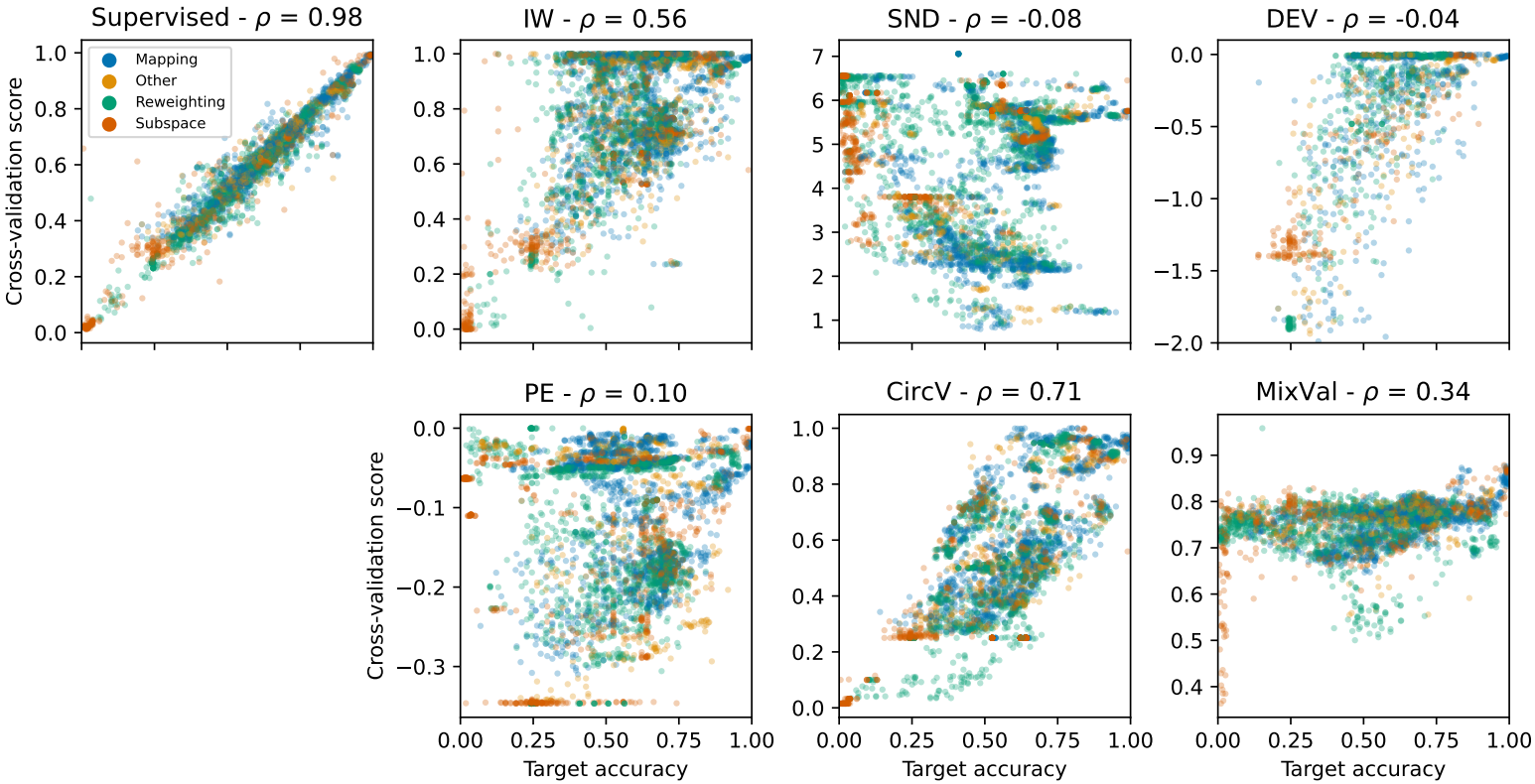
Adapt. helps, Adapt. hurts,, not statistically significant.

Benchmark results : the big table

		Cov. shift	Tar. shift	Cond. shift	Sub. shift	Office31	OfficeHome	MNIST/USPS	20NewsGroups	AmazonReview	Mushrooms	Phishing	BCI	Selected Scorer	Rank
	Train Src	0.88	0.85	0.66	0.19	0.65	0.56	0.54	0.59	0.7	0.72	0.91	0.55		10.66
	Train Tgt	0.92	0.93	0.82	0.98	0.89	0.8	0.96	1.0	0.73	1.0	0.97	0.64		1.55
Reweighting	Dens. RW	0.88	0.86	0.66	0.18	0.62	0.56	0.54	0.58	0.7	0.71	0.91	0.55	IW	12.20
	Disc. RW	0.85	0.83	0.71	0.18	0.63	0.54	0.5	0.6	0.68	0.75	0.91	0.56	CircV	8.75
	Gauss. RW	0.89	0.86	0.65	0.21	0.22	0.44	0.11	0.54	0.55	0.51	0.46	0.25	CircV	16.45
	KLIEP	0.88	0.86	0.66	0.19	0.65	0.56	0.54	0.6	0.69	0.72	0.91	0.55	CircV	10.56
	KMM	0.89	0.85	0.64	0.16	0.64	0.54	0.52	0.7	0.57	0.74	0.91	0.52	CircV	11.74
	NN RW	0.89	0.86	0.67	0.15	0.65	0.55	0.54	0.59	0.66	0.71	0.91	0.54	CircV	9.15
	MMDTarS	0.88	0.86	0.64	0.2	0.6	0.56	0.54	0.59	0.7	0.74	0.91	0.55	IW	10.81
Mapping	CORAL	0.74	0.7	0.76	0.18	0.65	0.57	0.62	0.73	0.7	0.72	0.92	0.62	CircV	5.08
	MapOT	0.72	0.57	0.82	0.02	0.6	0.51	0.61	0.76	0.68	0.63	0.84	0.47	PE	10.21
	EntOT	0.71	0.6	0.82	0.12	0.64	0.58	0.6	0.83	0.62	0.75	0.86	0.54	CircV	9.40
	ClassRegOT	0.74	0.58	0.81	0.11	NA	0.53	0.62	0.97	0.68	0.82	0.89	0.52	IW	8.25
	LinOT	0.73	0.73	0.76	0.18	0.66	0.57	0.64	0.82	0.7	0.76	0.91	0.61	CircV	4.06
	MMD-LS	0.78	0.72	0.76	0.56	0.65	0.56	0.55	0.97	0.63	0.85	NA	0.5	MixVal	8.22
Subspace	JPCA	0.88	0.85	0.66	0.15	0.62	0.48	0.51	0.77	0.69	0.78	0.9	0.54	PE	8.98
	SA	0.74	0.68	0.8	0.11	0.65	0.57	0.56	0.88	0.67	0.78	0.89	0.53	CircV	7.80
	TCA	0.52	0.47	0.51	0.62	0.04	0.02	0.07	0.61	0.61	0.49	0.48	0.26	DEV	17.58
	TSL	0.88	0.85	0.66	0.2	0.63	0.48	0.45	0.63	0.69	0.45	0.89	0.26	IW	15.09
Other	JDOT	0.72	0.58	0.82	0.13	0.6	0.42	0.59	0.79	0.67	0.65	0.79	0.47	IW	11.42
	OTLabelProp	0.72	0.59	0.8	0.07	0.66	0.56	0.62	0.86	0.67	0.64	0.86	0.5	CircV	10.01
	DASVM	0.89	0.86	0.65	0.15	NA	NA	NA	0.87	NA	0.83	0.85	NA	MixVal	7.29

Adapt. helps, Adapt. hurts,, not statistically significant.

Benchmark result: DA scorers



Compared DA scorers (CV score vs Target accuracy)

- IW: Importance Weighted [Sugiyama et al., 2007] and DEV [You et al., 2019].
- CircV: Circular Validation [Bruzzone and Marconcini, 2010].
- PE: Prediction entropy [Morerio et al., 2017].
- MixVal [Hu et al., 2023] and SND [Saito et al., 2021].

Not compared but at CAP 2025: MANO [Xie et al., 2024]

What about Deep DA?

	MNIST/USPS	Office31	OfficeHome	BCI	Selected Scorer	Rank
Train Src	0.85	0.77	0.58	0.54		6.19
Train Tgt	0.98	0.96	0.83	0.56		2.07
DeepCORAL [Sun and Saenko, 2016]	0.93	0.77	0.59	0.54	MixVal	3.29
DAN [Long et al., 2015]	0.86	0.75	0.56	0.53	IW	4.76
DANN [Ganin et al., 2016]	0.9	0.79	0.59	0.41	MixVal	4.98
DeepJDOT [Damodaran et al., 2018]	0.9	0.82	0.62	0.54	PE	2.92
MCC [Jin et al., 2020]	0.93	0.83	0.66	0.53	MixVal	2.38
MDD [Zhang et al., 2019]	0.87	0.78	0.56	0.4	MixVal	4.96
SPA [Xiao et al., 2023]	0.91	0.78	0.56	0.41	DEV	5.39
LinOT [Flamary et al., 2019]	0.64	0.6	0.57	0.61	CircV	

Deep learning benchmark results

- Same architecture as pre-trained feature extraction.
- Nested cross-validation to select parameters (same loop as shallow methods).
- On CV dataset, performance gains wrt shallow but clearly below what is published in the literature (as already seen in [Musgrave et al., 2021]).
- Perf. on biomedical data (BCI) is not as good and clearly below shallow methods.

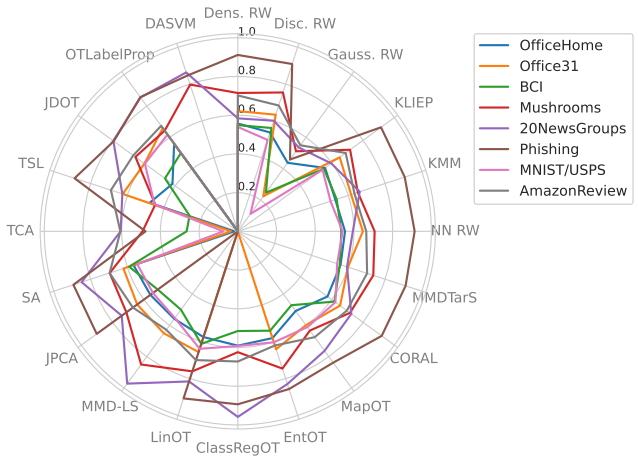
What about Deep DA?

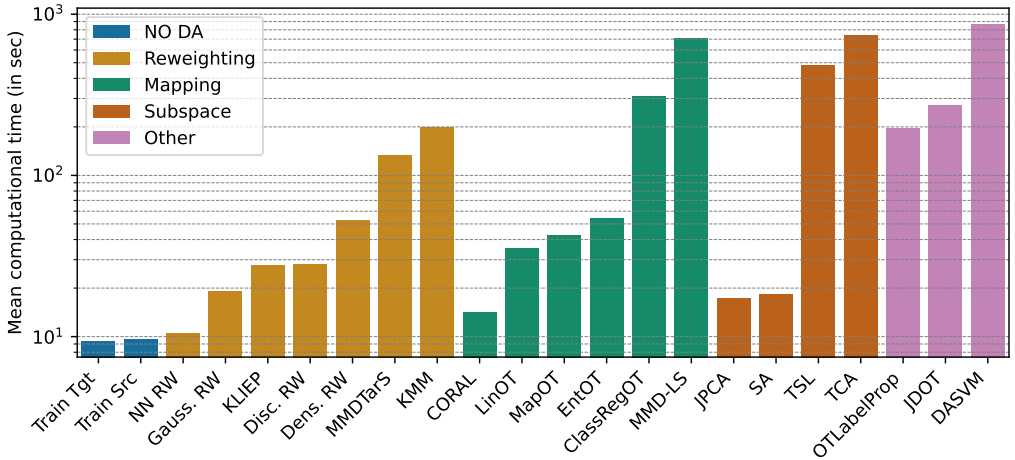
	MNIST/USPS	Office31	OfficeHome	BCI	Selected Scorer	Rank
Train Src	0.85	0.77	0.58	0.54		6.19
Train Tgt	0.98	0.96	0.83	0.56		2.07
DeepCORAL [Sun and Saenko, 2016]	0.93	0.77	0.59	0.54	MixVal	3.29
DAN [Long et al., 2015]	0.86	0.75	0.56	0.53	IW	4.76
DANN [Ganin et al., 2016]	0.9	0.79	0.59	0.41	MixVal	4.98
DeepJDOT [Damodaran et al., 2018]	0.9	0.82	0.62	0.54	PE	2.92
MCC [Jin et al., 2020]	0.93	0.83	0.66	0.53	MixVal	2.38
MDD [Zhang et al., 2019]	0.87	0.78	0.56	0.4	MixVal	4.96
SPA [Xiao et al., 2023]	0.91	0.78	0.56	0.41	DEV	5.39
LinOT [Flamary et al., 2019]	0.64	0.6	0.57	0.61	CircV	

Deep learning benchmark results

- Same architecture as pre-trained feature extraction.
- Nested cross-validation to select parameters (same loop as shallow methods).
- On CV dataset, performance gains wrt shallow but clearly below what is published in the literature (as already seen in [Musgrave et al., 2021]).
- Perf. on biomedical data (BCI) is not as good and clearly below shallow methods.

SKADA-bench: conclusions





Conclusions from results [Lalou et al., 2025]

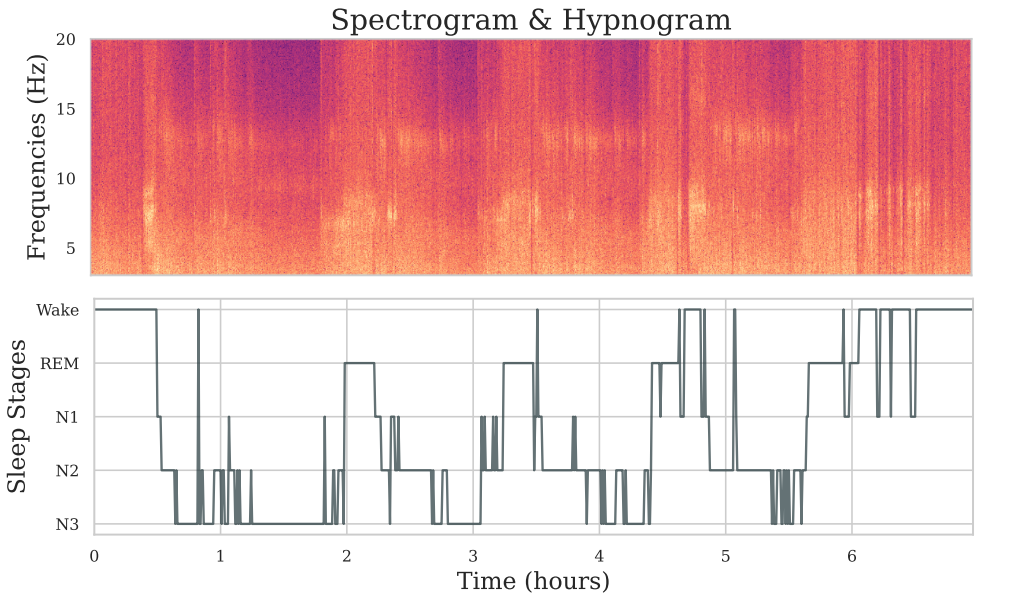
- DA is a complex task, no single method works best for all datasets.
- Simple linear mapping methods (LinearOT, CORAL) often help.
- Robust DA scorers are Circular Validation and Importance Weighting.
- Computational complexity depends on the method and its number of parameters.
- On specific modalities with large datasets, deep DA methods can work very well.
- Can we have the best of both shallow and deep worlds?

Domain Adaptation for Sleep Staging

Sleep stage classification from EEG signals

Subject S_1

(X_1, y_1)

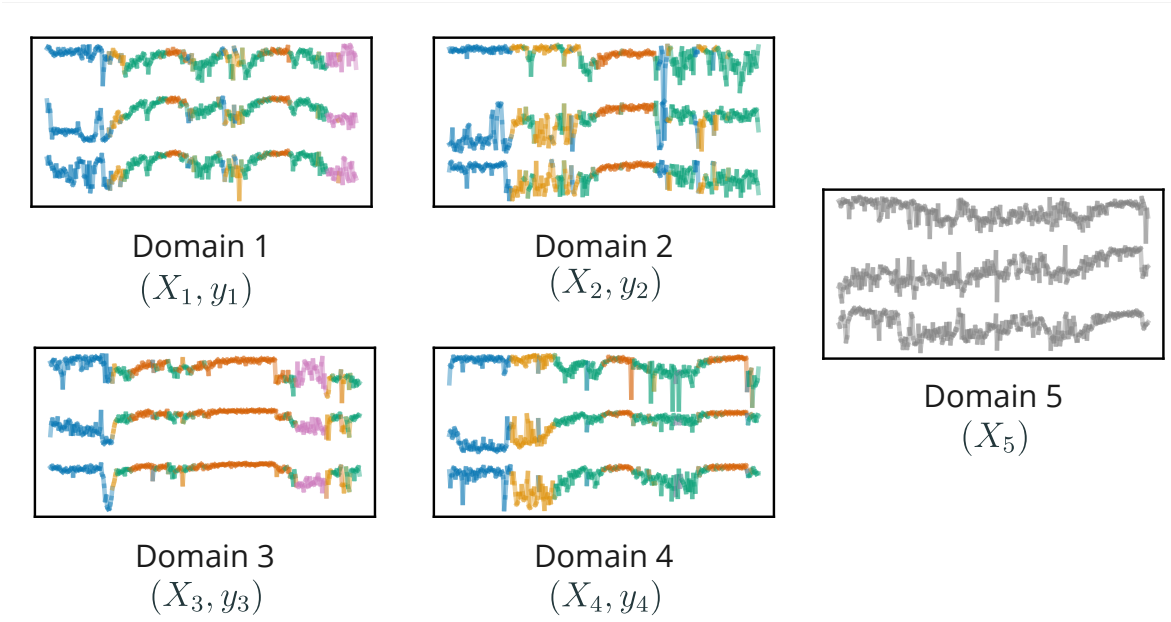


Challenge

- **Objective:** Classify sleep stages (5 classes) from EEG signals of night recordings.
- **Source data:** Labeled EEG signals from multiple subjects/hospitals.
- **Target data:** EEG signals from a new subject/hospital without labels.
- **Data shift:** between all the domains due to (sensor, setup, subject, hospital, etc.)

$\rightarrow$ Train a DNN model that can adapt to the target data at test time (no retrain).

Sleep stage classification from EEG signals

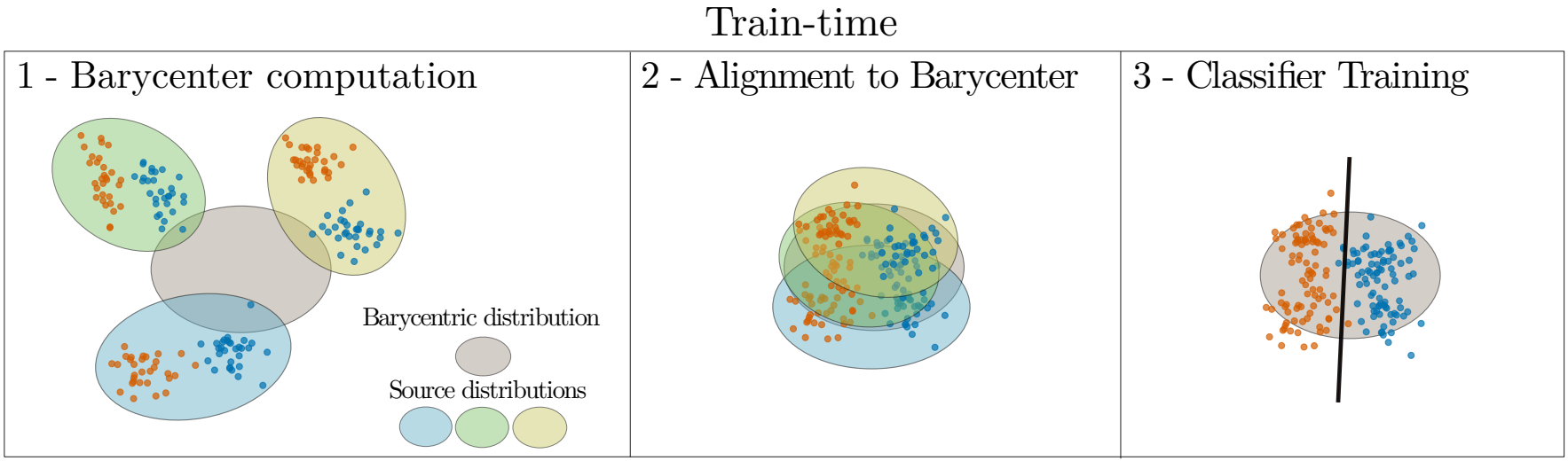


Challenge

- **Objective:** Classify sleep stages (5 classes) from EEG signals of night recordings.
- **Source data:** Labeled EEG signals from multiple subjects/hospitals.
- **Target data:** EEG signals from a new subject/hospital without labels.
- **Data shift:** between all the domains due to (sensor, setup, subject, hospital, etc.)

$\rightarrow$ Train a DNN model that can adapt to the target data at test time (no retrain).

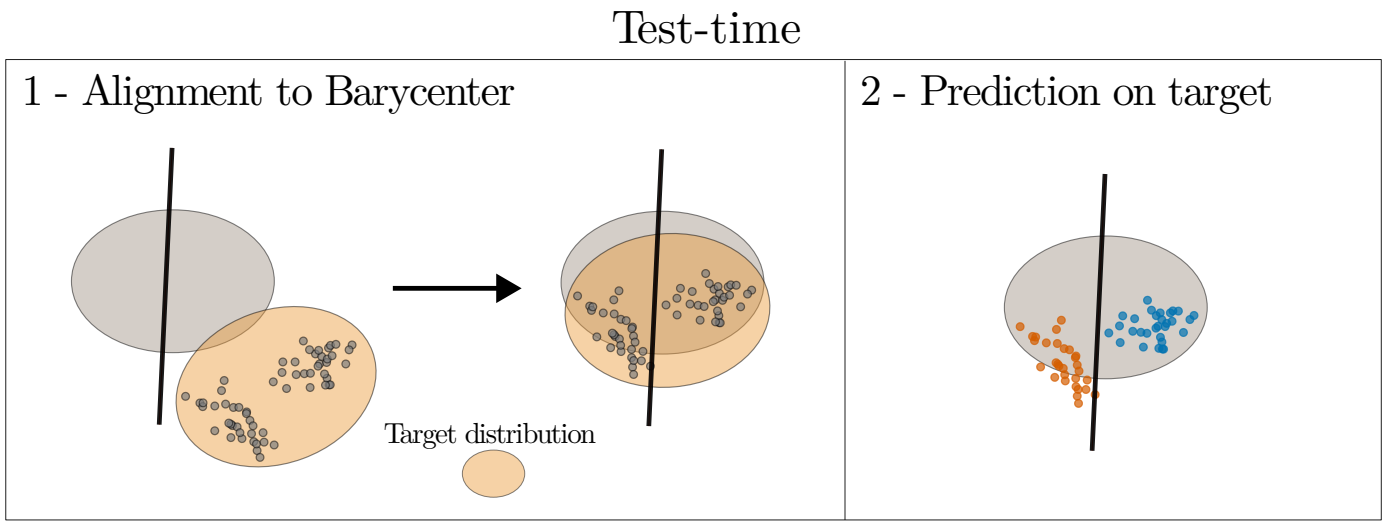
Convolutional Monge Mapping Normalization



Principle [Gnassounou et al., 2023]

- Main idea: adapt with optimal transport all domains to an "average" domain to remove domain shifts [Montesuma and Mboula, 2021].
- Train time:
  1. Estimate barycenter of all source domains.
  2. Map each source domain to the barycenter with a Monge mapping.
  3. Train a classifier on the barycenter.
- Test time:
  1. Map the target domain to the barycenter with the Monge mapping.
  2. Predict with the predictor trained on the barycenter.
- Use Linear OT on signals using Fourier transform for fast computation.

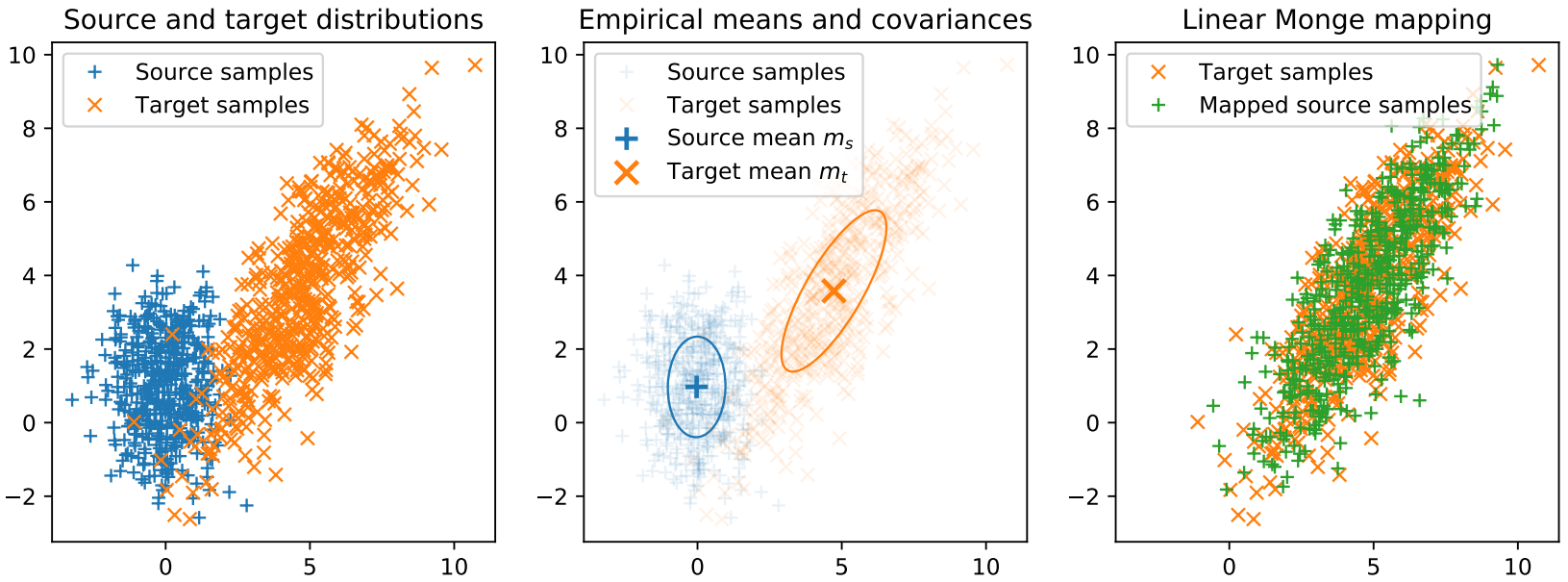
Convolutional Monge Mapping Normalization



Principle [Gnassounou et al., 2023]

- Main idea: adapt with optimal transport all domains to an "average" domain to remove domain shifts [Montesuma and Mboula, 2021].
- Train time:
  1. Estimate barycenter of all source domains.
  2. Map each source domain to the barycenter with a Monge mapping.
  3. Train a classifier on the barycenter.
- Test time:
  1. Map the target domain to the barycenter with the Monge mapping.
  2. Predict with the predictor trained on the barycenter.
- Use Linear OT on signals using Fourier transform for fast computation.

Optimal Transport between Gaussians



OT mapping between Gaussian distributions

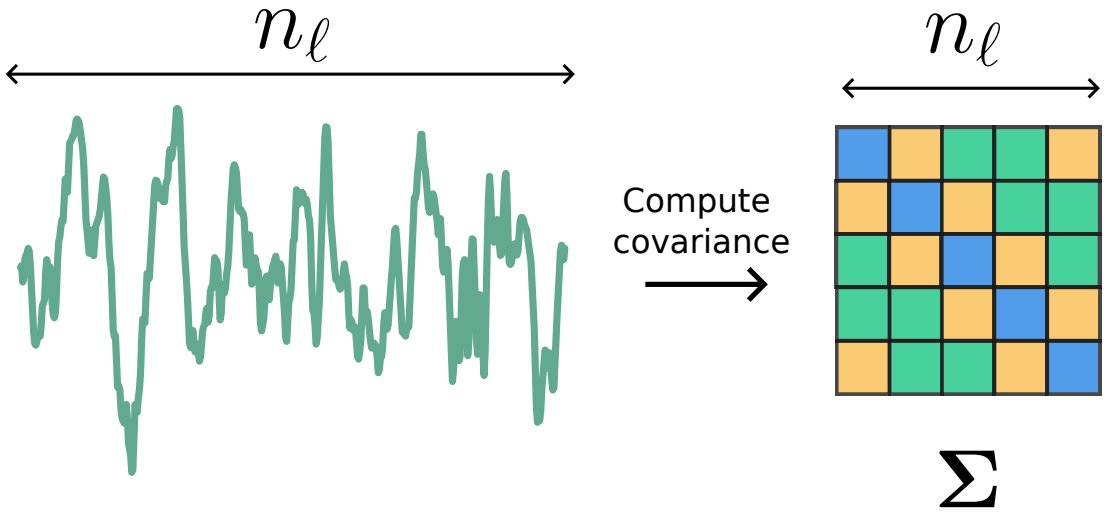
- $\mu_s \sim \mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1)$ and $\mu_t \sim \mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)$
- The optimal map m for $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$ is given by

$$m(\mathbf{x}) = \mathbf{m}_2 + A(\mathbf{x} - \mathbf{m}_1) \quad \text{with} \quad A = \mathbf{\Sigma}_1^{-1/2}(\mathbf{\Sigma}_1^{1/2}\mathbf{\Sigma}_2\mathbf{\Sigma}_1^{1/2})^{1/2}\mathbf{\Sigma}_1^{-1/2}$$

- The barycenter between Gaussians $\mu_k \sim \mathcal{N}(\mathbf{m}_k, \Sigma_k)$ is computed with:

$$\bar{\mathbf{m}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{m}_k \ , \quad \bar{\mathbf{\Sigma}} = \frac{1}{K}\sum_{k=1}^{K} \left(\bar{\mathbf{\Sigma}}^{\frac{1}{2}}\mathbf{\Sigma}_k\bar{\mathbf{\Sigma}}^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

Gaussianity assumption for signals



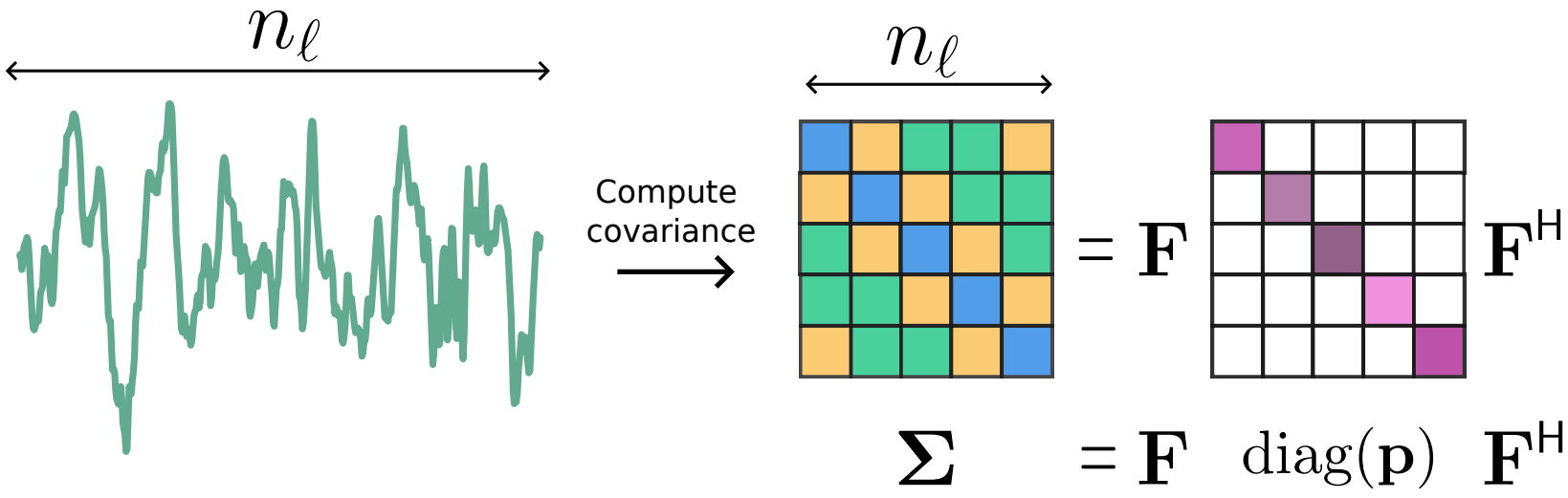
Stationary and periodic signals Gaussian signal

- Centered Gaussian distributions $\rightarrow \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma} \in \mathcal{S}_{n_\ell}^{++}$
- $\mathbf{\Sigma}$ is the "auto-covariance", computed with time-lagged. $\mathbf{\Sigma}_{i,j} = \mathbf{X}_i^\top \mathbf{X}_j$
- Stationarity+Periodicity $\rightarrow$ Cov. matrices are Toeplitz circulant matrices.
- The Discrete Fourier Transform (DFT) can diagonalize the circulant matrix

$$\mathbf{\Sigma} = \mathbf{F}\text{diag}(\mathbf{p})\mathbf{F}^H \ ,$$

with $\mathbf{F}$ Fourier transform operator and $\mathbf{p}$ the Power Spectral Density (PSD).

Gaussianity assumption for signals



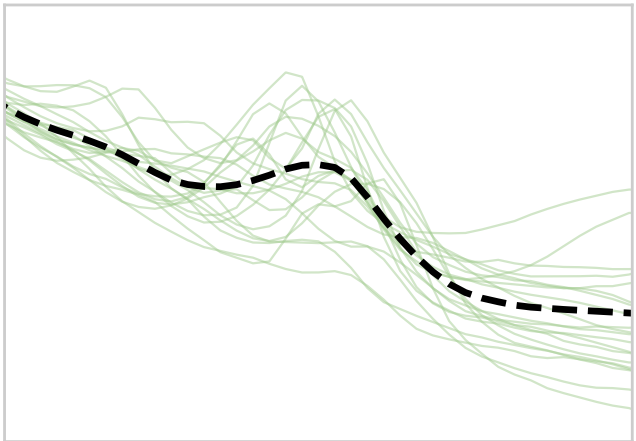
Stationary and periodic signals Gaussian signal

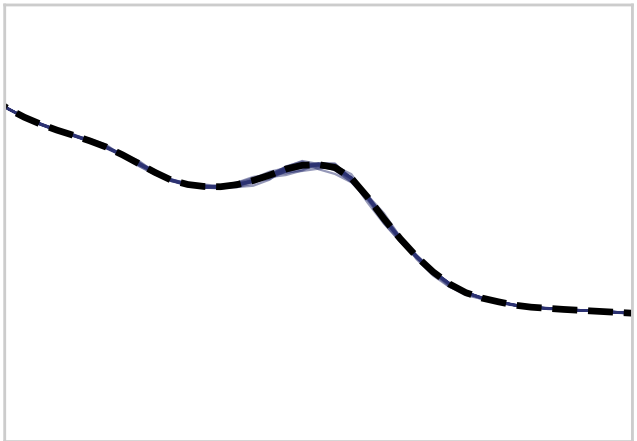
- Centered Gaussian distributions $\rightarrow \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma} \in \mathcal{S}_{n_\ell}^{++}$
- $\mathbf{\Sigma}$ is the "auto-covariance", computed with time-lagged. $\mathbf{\Sigma}_{i,j} = \mathbf{X}_i^\top \mathbf{X}_j$
- Stationarity+Periodicity $\rightarrow$ Cov. matrices are Toeplitz circulant matrices.
- The Discrete Fourier Transform (DFT) can diagonalize the circulant matrix

$$\mathbf{\Sigma} = \mathbf{F}\mathrm{diag}(\mathbf{p})\mathbf{F}^H \ ,$$

with $\mathbf{F}$ Fourier transform operator and $\mathbf{p}$ the Power Spectral Density (PSD).

Convolutional Monge Mapping Normalization (CMMN)





CMMN principle [Gnassounou et al., 2023]

- Estimate the Power Spectral Density (PSD) $\mathbf{p}_k$ of each signal with Welch method.
- Compute the barycenter of the PSDs with the closed form

$$\bar{\mathbf{p}} = \left(\frac{1}{K} \sum_{k=1}^{K} \mathbf{p}_k^{\odot \frac{1}{2}} \right)^{\odot 2} .$$

Barycenter PSD — Domain k PSD

- Compute the convolutional mapping from each domain to the barycenter with

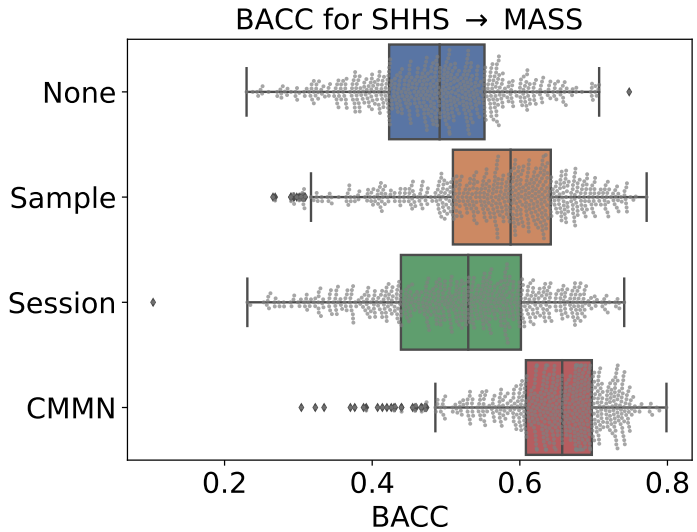
$$m_k(\mathbf{x}) = \mathbf{h}_k * \mathbf{x} , \quad \text{with} \quad \mathbf{h}_k = \mathbf{F}^H \left(\bar{\mathbf{p}}^{\odot \frac{1}{2}} \odot \mathbf{p}_k^{\odot -\frac{1}{2}} \right) .$$

Monge filter — Barycenter PSD — Domain k PSD

- Train a deep learning predictor on the mapped signals (apply on mapped target).

CMMN numerical experiments

	No Adapt	CMMN
MASS→MASS	75.1 ± 1.0	$\mathbf{76.2 \pm 2.2}$
Phys.→Phys.	69.2 ± 2.7	$\mathbf{71.7 \pm 2.4}$
SHHS→SHHS	61.2 ± 3.8	$\mathbf{64.3 \pm 2.7}$
MASS→Phys.	58.4 ± 2.4	$\mathbf{62.3 \pm 1.5}$
MASS→SHHS	41.8 ± 3.6	$\mathbf{47.6 \pm 4.0}$
Phys.→MASS	64.0 ± 2.7	$\mathbf{68.3 \pm 2.5}$
Phys.→SHHS	45.6 ± 2.1	$\mathbf{51.6 \pm 1.8}$
SHHS→MASS	57.0 ± 2.8	$\mathbf{64.5 \pm 2.8}$
SHHS→Phys.	55.0 ± 2.7	$\mathbf{58.3 \pm 1.7}$
Mean	58.6 ± 2.6	$\mathbf{62.7 \pm 2.4}$

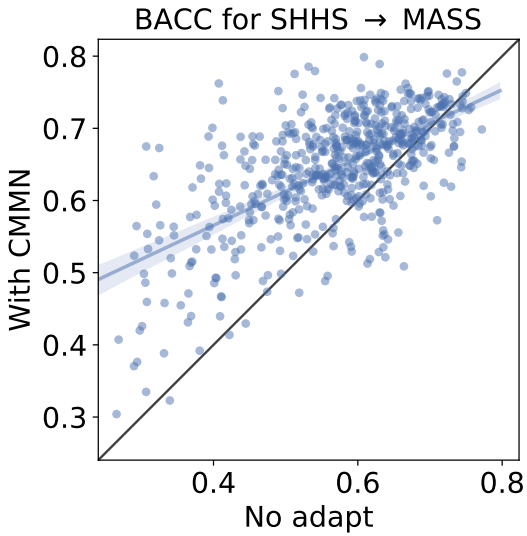


Setup and results

- Experiment on 3 datasets (MASS, Physionet and SHHS).
- Use the neural network architecture and setup from [Chambon et al., 2018].
- Compute performance across datasets with Balanced Accuracy (BAC).
- CMMN improves the BAC in average but especially on subjects with poor performance without adaptation.
- Can be extendes to multivariate signals [Gnassounou et al., 2024].
- Limitation: few datasets, arch. not SOTA, **CMMN is linear only pre-processing.**

CMMN numerical experiments

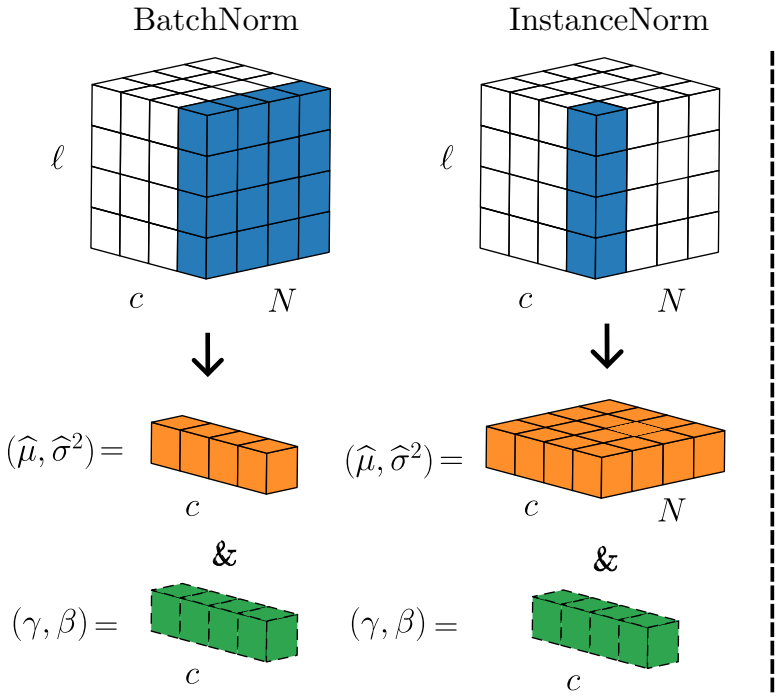
	No Adapt	CMMN
MASS→MASS	75.1 ± 1.0	$\mathbf{76.2 \pm 2.2}$
Phys.→Phys.	69.2 ± 2.7	$\mathbf{71.7 \pm 2.4}$
SHHS→SHHS	61.2 ± 3.8	$\mathbf{64.3 \pm 2.7}$
MASS→Phys.	58.4 ± 2.4	$\mathbf{62.3 \pm 1.5}$
MASS→SHHS	41.8 ± 3.6	$\mathbf{47.6 \pm 4.0}$
Phys.→MASS	64.0 ± 2.7	$\mathbf{68.3 \pm 2.5}$
Phys.→SHHS	45.6 ± 2.1	$\mathbf{51.6 \pm 1.8}$
SHHS→MASS	57.0 ± 2.8	$\mathbf{64.5 \pm 2.8}$
SHHS→Phys.	55.0 ± 2.7	$\mathbf{58.3 \pm 1.7}$
Mean	58.6 ± 2.6	$\mathbf{62.7 \pm 2.4}$

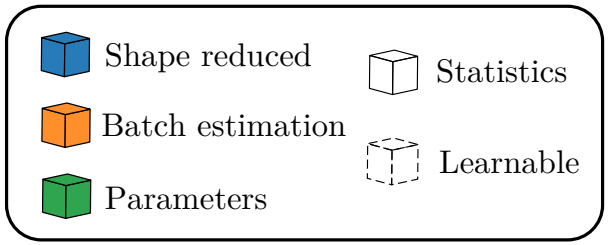


Setup and results

- Experiment on 3 datasets (MASS, Physionet and SHHS).
- Use the neural network architecture and setup from [Chambon et al., 2018].
- Compute performance across datasets with Balanced Accuracy (BAC).
- CMMN improves the BAC in average but especially on subjects with poor performance without adaptation.
- Can be extendes to multivariate signals [Gnassounou et al., 2024].
- Limitation: few datasets, arch. not SOTA, **CMMN is linear only pre-processing.**

PSDNorm as a new normalization layer

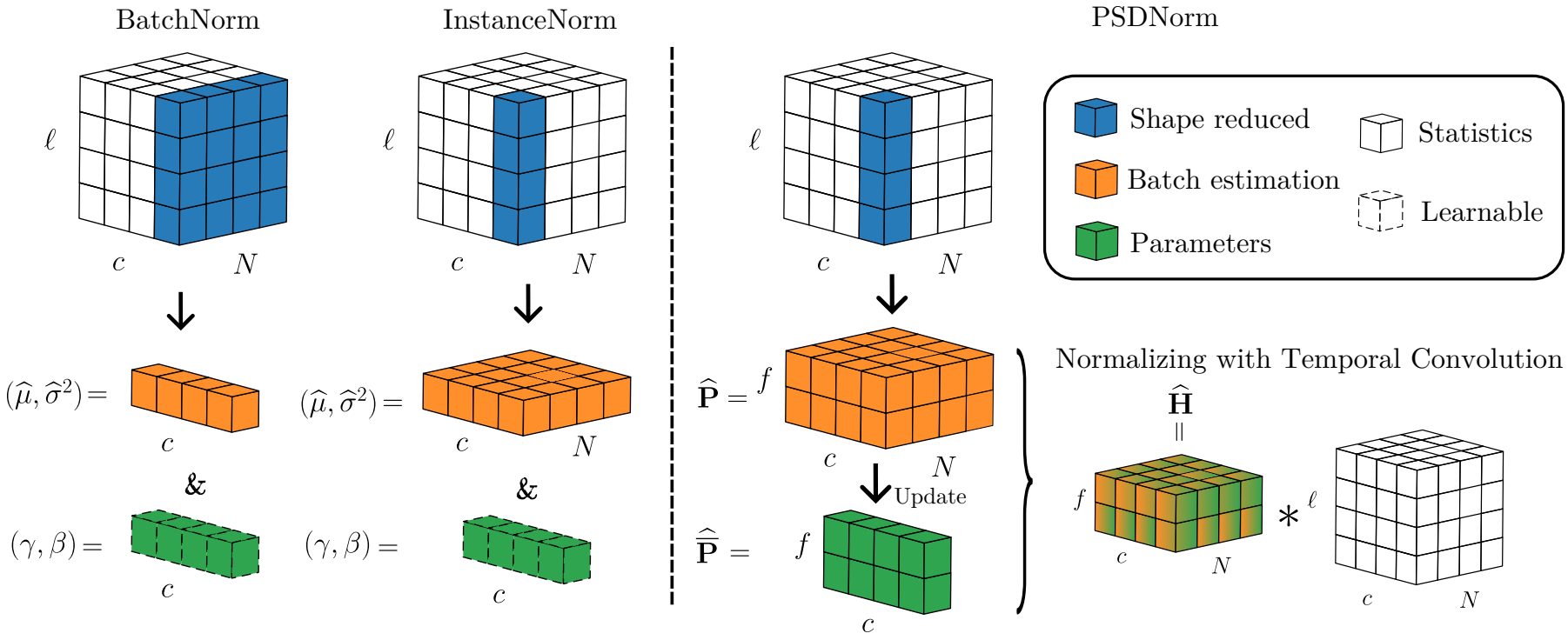




Principle [Gnassounou et al., 2025]

- Standard way to reduce variability in DL is to use Normalization layers.
- PSDNorm is a new normalization layer that takes into account the temporal correlation of the signals (through its PSD).
- Conv. Monge map. with barycenter estimated with geodesic update on the batch.
- Can be used as replacement of other normalization in any deep learning architecture.

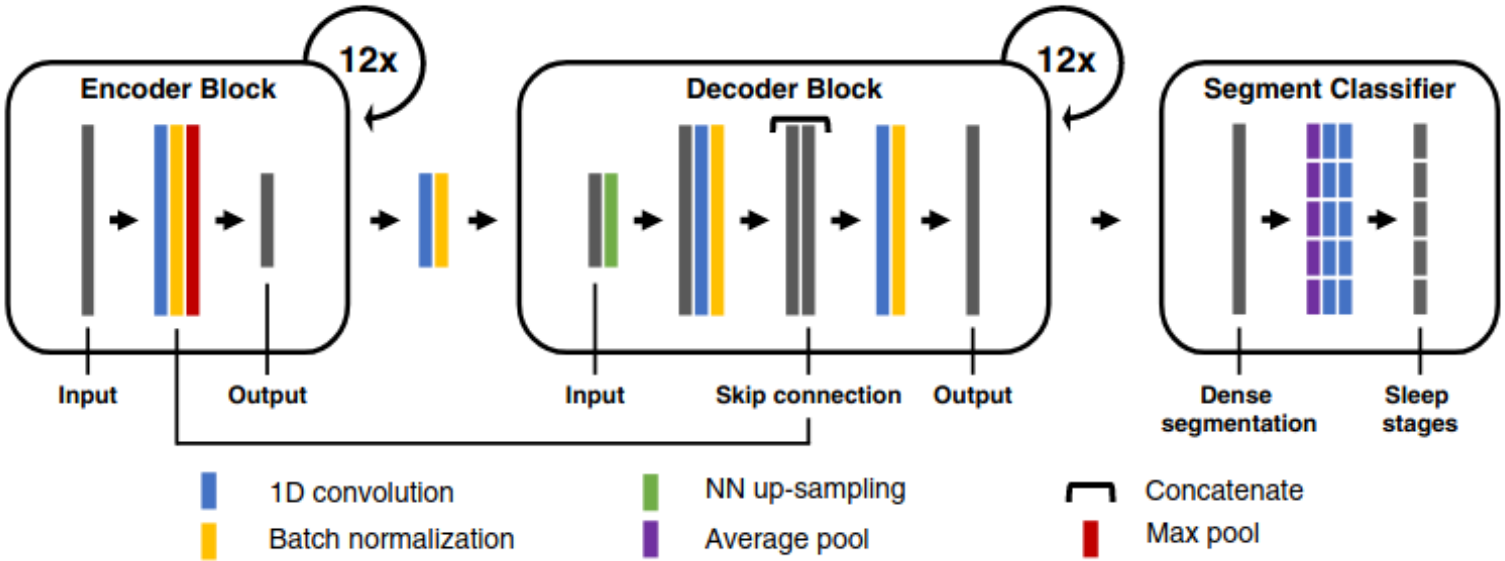
PSDNorm as a new normalization layer



Principle [Gnassounou et al., 2025]

- Standard way to reduce variability in DL is to use Normalization layers.
- PSDNorm is a new normalization layer that takes into account the temporal correlation of the signals (through its PSD).
- Conv. Monge map. with barycenter estimated with geodesic update on the batch.
- Can be used as replacement of other normalization in any deep learning architecture.

Numerical experiments for PSDNorm



Experimental setup

- Use the Usleep architecture [Perslev et al., 2021] based on UNet.
- Take as input sequence of 17min of sleep (one annotation every 30 seconds)
- Data: 10k subjects, 10M of samples, 350 Go on disk.
- Perform Leave-One-Dataset-Out (LODO) cross-validation on all datasets.
- Thank you Jean Zay for the GPU compute.

Results on LODO sleep staging

		BatchNorm	LayerNorm	InstanceNorm	PSDNorm(F=5)	PSDNorm(F=9)	PSDNorm(F=17)
All subjects	ABC	78.49 ± 0.42	77.94 ± 0.31	**78.83 ± 0.59**	78.56 ± 0.67	78.73 ± 0.32	78.60 ± 0.28
	CCSHS	**88.79 ± 0.21**	87.51 ± 0.77	88.75 ± 0.04	88.56 ± 0.36	88.48 ± 0.07	88.52 ± 0.19
	CFS	84.97 ± 0.37	84.29 ± 0.67	**85.73 ± 0.29**	85.42 ± 0.09	85.25 ± 0.19	85.28 ± 0.12
	CHAT	64.72 ± 3.94	64.36 ± 0.40	68.86 ± 2.49	70.57 ± 1.24	70.36 ± 2.22	**70.71 ± 2.15**
	HOMEPAP	76.39 ± 0.29	75.23 ± 0.78	76.70 ± 0.35	76.72 ± 0.27	76.93 ± 0.10	**77.02 ± 0.32**
	MASS	**73.71 ± 0.62**	71.39 ± 3.00	72.12 ± 0.70	72.51 ± 1.68	71.34 ± 2.68	71.61 ± 0.71
	MROS	81.30 ± 0.25	80.44 ± 0.29	81.49 ± 0.18	**81.57 ± 0.34**	81.35 ± 0.17	81.30 ± 0.13
	PhysioNet	76.13 ± 0.57	75.12 ± 0.22	76.15 ± 0.52	75.96 ± 1.02	**76.35 ± 0.27**	76.02 ± 0.47
	SHHS	77.97 ± 1.46	75.98 ± 0.48	79.05 ± 0.89	79.14 ± 1.01	**79.33 ± 0.62**	79.12 ± 0.11
	SOF	81.33 ± 0.54	81.82 ± 0.79	81.98 ± 0.22	**82.50 ± 0.34**	82.15 ± 1.00	81.94 ± 0.65
	Mean	78.38 ± 0.47	77.41 ± 0.28	78.97 ± 0.11	**79.15 ± 0.14**	79.03 ± 0.10	79.01 ± 0.36
Balanced@400	ABC	78.26 ± 1.33	75.29 ± 0.81	**78.73 ± 0.42**	78.18 ± 0.68	78.18 ± 0.91	77.76 ± 1.00
	CCSHS	87.42 ± 0.16	85.20 ± 0.48	**87.62 ± 0.42**	87.58 ± 0.30	87.35 ± 0.52	**87.62 ± 0.48**
	CFS	84.32 ± 0.57	81.66 ± 1.36	**84.72 ± 0.33**	84.29 ± 0.36	84.06 ± 0.10	84.46 ± 0.06
	CHAT	66.55 ± 0.88	61.19 ± 1.16	64.43 ± 4.41	**70.28 ± 1.70**	68.11 ± 3.94	69.88 ± 0.46
	HOMEPAP	75.25 ± 0.50	74.86 ± 0.25	76.47 ± 0.63	**76.83 ± 0.61**	76.61 ± 0.74	76.49 ± 0.45
	MASS	70.00 ± 1.91	68.56 ± 3.33	71.52 ± 1.13	72.77 ± 1.09	**73.07 ± 1.30**	72.23 ± 2.40
	MROS	80.37 ± 0.20	78.05 ± 0.22	80.28 ± 0.21	80.26 ± 0.11	80.32 ± 0.22	**80.70 ± 0.42**
	PhysioNet	**75.81 ± 0.13**	71.82 ± 2.12	74.68 ± 0.55	74.82 ± 2.11	73.77 ± 1.73	75.09 ± 0.97
	SHHS	76.44 ± 0.92	75.12 ± 0.39	78.68 ± 0.37	**78.88 ± 0.68**	77.28 ± 0.91	78.41 ± 0.49
	SOF	81.08 ± 1.14	78.70 ± 0.50	80.68 ± 1.38	79.49 ± 0.41	**81.44 ± 0.97**	81.07 ± 0.66
	Mean	77.55 ± 0.34	75.05 ± 0.28	77.78 ± 0.46	78.34 ± 0.42	78.02 ± 0.67	**78.37 ± 0.38**

Results on the left-out datasets

- F is size of Welsh PSD estimator and convolutuional filter.
- Balanced@400 setting: use 400 subjects per dataset ($\sim$10% of the subjects)
- PSDNorm is the best normalization in both settings (but needs less data).

Results on LODO sleep staging

		BatchNorm	LayerNorm	InstanceNorm	PSDNorm(F=5)	PSDNorm(F=9)	PSDNorm(F=17)
All subjects	ABC	78.49 ± 0.42	77.94 ± 0.31	**78.83 ± 0.59**	78.56 ± 0.67	78.73 ± 0.32	78.60 ± 0.28
	CCSHS	**88.79 ± 0.21**	87.51 ± 0.77	88.75 ± 0.04	88.56 ± 0.36	88.48 ± 0.07	88.52 ± 0.19
	CFS	84.97 ± 0.37	84.29 ± 0.67	**85.73 ± 0.29**	85.42 ± 0.09	85.25 ± 0.19	85.28 ± 0.12
	CHAT	64.72 ± 3.94	64.36 ± 0.40	68.86 ± 2.49	70.57 ± 1.24	70.36 ± 2.22	**70.71 ± 2.15**
	HOMEPAP	76.39 ± 0.29	75.23 ± 0.78	76.70 ± 0.35	76.72 ± 0.27	76.93 ± 0.10	**77.02 ± 0.32**
	MASS	**73.71 ± 0.62**	71.39 ± 3.00	72.12 ± 0.70	72.51 ± 1.68	71.34 ± 2.68	71.61 ± 0.71
	MROS	81.30 ± 0.25	80.44 ± 0.29	81.49 ± 0.18	**81.57 ± 0.34**	81.35 ± 0.17	81.30 ± 0.13
	PhysioNet	76.13 ± 0.57	75.12 ± 0.22	76.15 ± 0.52	75.96 ± 1.02	**76.35 ± 0.27**	76.02 ± 0.47
	SHHS	77.97 ± 1.46	75.98 ± 0.48	79.05 ± 0.89	79.14 ± 1.01	**79.33 ± 0.62**	79.12 ± 0.11
	SOF	81.33 ± 0.54	81.82 ± 0.79	81.98 ± 0.22	**82.50 ± 0.34**	82.15 ± 1.00	81.94 ± 0.65
	Mean	78.38 ± 0.47	77.41 ± 0.28	78.97 ± 0.11	**79.15 ± 0.14**	79.03 ± 0.10	79.01 ± 0.36
Balanced@400	ABC	78.26 ± 1.33	75.29 ± 0.81	**78.73 ± 0.42**	78.18 ± 0.68	78.18 ± 0.91	77.76 ± 1.00
	CCSHS	87.42 ± 0.16	85.20 ± 0.48	**87.62 ± 0.42**	87.58 ± 0.30	87.35 ± 0.52	**87.62 ± 0.48**
	CFS	84.32 ± 0.57	81.66 ± 1.36	**84.72 ± 0.33**	84.29 ± 0.36	84.06 ± 0.10	84.46 ± 0.06
	CHAT	66.55 ± 0.88	61.19 ± 1.16	64.43 ± 4.41	**70.28 ± 1.70**	68.11 ± 3.94	69.88 ± 0.46
	HOMEPAP	75.25 ± 0.50	74.86 ± 0.25	76.47 ± 0.63	**76.83 ± 0.61**	76.61 ± 0.74	76.49 ± 0.45
	MASS	70.00 ± 1.91	68.56 ± 3.33	71.52 ± 1.13	72.77 ± 1.09	**73.07 ± 1.30**	72.23 ± 2.40
	MROS	80.37 ± 0.20	78.05 ± 0.22	80.28 ± 0.21	80.26 ± 0.11	80.32 ± 0.22	**80.70 ± 0.42**
	PhysioNet	**75.81 ± 0.13**	71.82 ± 2.12	74.68 ± 0.55	74.82 ± 2.11	73.77 ± 1.73	75.09 ± 0.97
	SHHS	76.44 ± 0.92	75.12 ± 0.39	78.68 ± 0.37	**78.88 ± 0.68**	77.28 ± 0.91	78.41 ± 0.49
	SOF	81.08 ± 1.14	78.70 ± 0.50	80.68 ± 1.38	79.49 ± 0.41	**81.44 ± 0.97**	81.07 ± 0.66
	Mean	77.55 ± 0.34	75.05 ± 0.28	77.78 ± 0.46	78.34 ± 0.42	78.02 ± 0.67	**78.37 ± 0.38**

Results on the left-out datasets

- F is size of Welsh PSD estimator and convolutuional filter.
- Balanced@400 setting: use 400 subjects per dataset (∼10% of the subjects)
- PSDNorm is the best normalization in both settings (but needs less data).

Results on LODO sleep staging

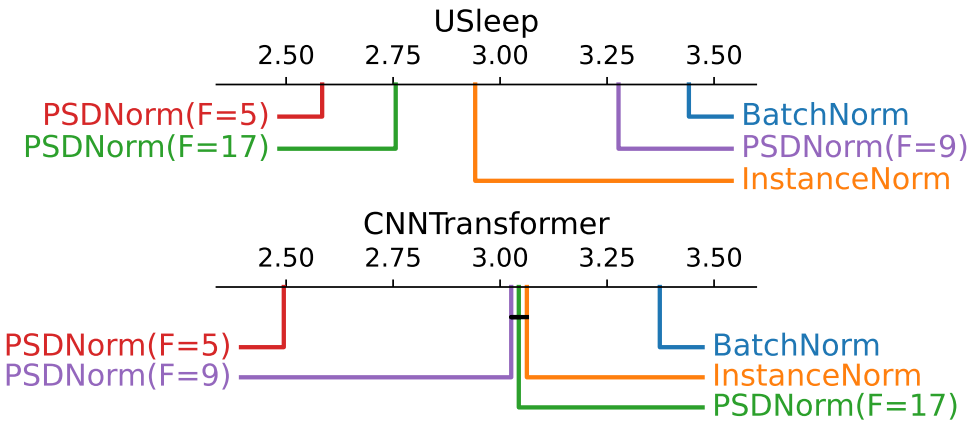
		BatchNorm	LayerNorm	InstanceNorm	PSDNorm(F=5)	PSDNorm(F=9)	PSDNorm(F=17)
All subjects	ABC	78.49 ± 0.42	77.94 ± 0.31	**78.83 ± 0.59**	78.56 ± 0.67	78.73 ± 0.32	78.60 ± 0.28
	CCSHS	**88.79 ± 0.21**	87.51 ± 0.77	88.75 ± 0.04	88.56 ± 0.36	88.48 ± 0.07	88.52 ± 0.19
	CFS	84.97 ± 0.37	84.29 ± 0.67	**85.73 ± 0.29**	85.42 ± 0.09	85.25 ± 0.19	85.28 ± 0.12
	CHAT	64.72 ± 3.94	64.36 ± 0.40	68.86 ± 2.49	70.57 ± 1.24	70.36 ± 2.22	**70.71 ± 2.15**
	HOMEPAP	76.39 ± 0.29	75.23 ± 0.78	76.70 ± 0.35	76.72 ± 0.27	76.93 ± 0.10	**77.02 ± 0.32**
	MASS	**73.71 ± 0.62**	71.39 ± 3.00	72.12 ± 0.70	72.51 ± 1.68	71.34 ± 2.68	71.61 ± 0.71
	MROS	81.30 ± 0.25	80.44 ± 0.29	81.49 ± 0.18	**81.57 ± 0.34**	81.35 ± 0.17	81.30 ± 0.13
	PhysioNet	76.13 ± 0.57	75.12 ± 0.22	76.15 ± 0.52	75.96 ± 1.02	**76.35 ± 0.27**	76.02 ± 0.47
	SHHS	77.97 ± 1.46	75.98 ± 0.48	79.05 ± 0.89	79.14 ± 1.01	**79.33 ± 0.62**	79.12 ± 0.11
	SOF	81.33 ± 0.54	81.82 ± 0.79	81.98 ± 0.22	**82.50 ± 0.34**	82.15 ± 1.00	81.94 ± 0.65
	Mean	78.38 ± 0.47	77.41 ± 0.28	78.97 ± 0.11	**79.15 ± 0.14**	79.03 ± 0.10	79.01 ± 0.36
Balanced@400	ABC	78.26 ± 1.33	75.29 ± 0.81	**78.73 ± 0.42**	78.18 ± 0.68	78.18 ± 0.91	77.76 ± 1.00
	CCSHS	87.42 ± 0.16	85.20 ± 0.48	**87.62 ± 0.42**	87.58 ± 0.30	87.35 ± 0.52	**87.62 ± 0.48**
	CFS	84.32 ± 0.57	81.66 ± 1.36	**84.72 ± 0.33**	84.29 ± 0.36	84.06 ± 0.10	84.46 ± 0.06
	CHAT	66.55 ± 0.88	61.19 ± 1.16	64.43 ± 4.41	**70.28 ± 1.70**	68.11 ± 3.94	69.88 ± 0.46
	HOMEPAP	75.25 ± 0.50	74.86 ± 0.25	76.47 ± 0.63	**76.83 ± 0.61**	76.61 ± 0.74	76.49 ± 0.45
	MASS	70.00 ± 1.91	68.56 ± 3.33	71.52 ± 1.13	72.77 ± 1.09	**73.07 ± 1.30**	72.23 ± 2.40
	MROS	80.37 ± 0.20	78.05 ± 0.22	80.28 ± 0.21	80.26 ± 0.11	80.32 ± 0.22	**80.70 ± 0.42**
	PhysioNet	**75.81 ± 0.13**	71.82 ± 2.12	74.68 ± 0.55	74.82 ± 2.11	73.77 ± 1.73	75.09 ± 0.97
	SHHS	76.44 ± 0.92	75.12 ± 0.39	78.68 ± 0.37	**78.88 ± 0.68**	77.28 ± 0.91	78.41 ± 0.49
	SOF	81.08 ± 1.14	78.70 ± 0.50	80.68 ± 1.38	79.49 ± 0.41	**81.44 ± 0.97**	81.07 ± 0.66
	Mean	77.55 ± 0.34	75.05 ± 0.28	77.78 ± 0.46	78.34 ± 0.42	78.02 ± 0.67	**78.37 ± 0.38**

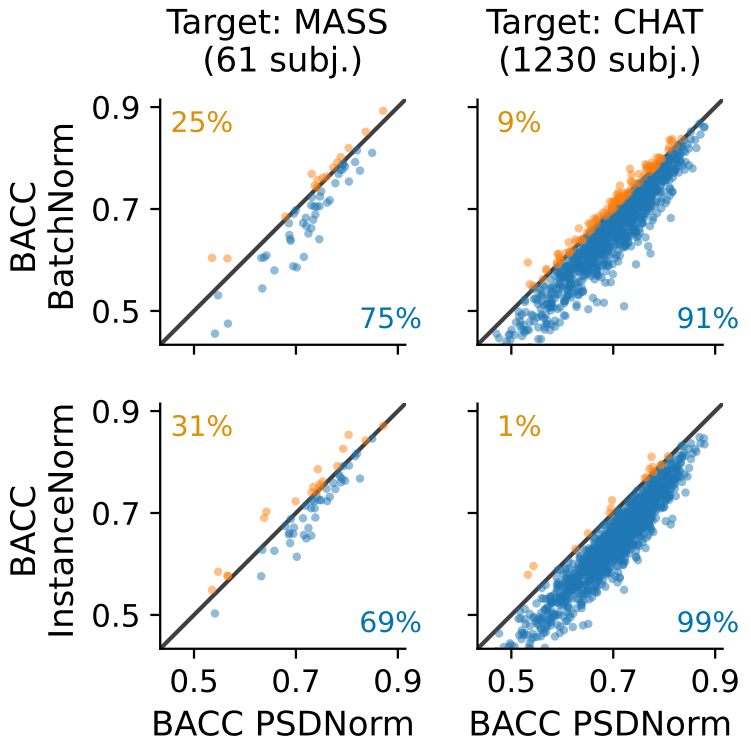
Results on the left-out datasets

- F is size of Welsh PSD estimator and convolutuional filter.
- Balanced@400 setting: use 400 subjects per dataset (∼10% of the subjects)
- PSDNorm is the best normalization in both settings (but needs less data).

Results: Comparison with other normalizations

Critical Difference Diagram for the Balanced@400 setting





Comparison results

- Critical Difference Diagram reports the average rank of each normalization method and test their statistical significance.
- PSDNorm is significantly better than other normalizations (on several archi.)
- PSDNorm consistently improve the subjects' performance especially for the most challenging ones in scatterplots.

Conclusion

Collaborators on Optimal Transport for Domain Adaptation


N. Courty


A. Rakotomamonjy


D. Tuia


A. Habrard


B. Damodaran


K. Lounici


A. Ferrari


K. Fatras

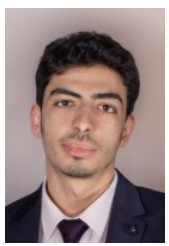

I. Redko


A. de Mathelin


A. Gramfort


T. Gnassounou


A. Collas


Y. Lalou


T. Moreau

+ All SKADA and SKADA-bench contributors

Domain Adaptation

- Domain Adaptation is a challenging problem in practice.
- Simple methods like CORAL, or LinearOT are easier to use and validate.
- User knowledge of the data is crucial to choose the right method or design it.
- Applied Deep DA requires to adapt both training (DA loss) and architecture.
- Théo Gnassounou is looking for a postdoc (open to research-domain adaptation)!

Thank you for your attention!

References i

[Bruzzone and Marconcini, 2010] Bruzzone, L. and Marconcini, M. (2010). **Domain adaptation problems: A dasvm classification technique and a circular validation strategy.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):770–787.

[Chambon et al., 2018] Chambon, S., Galtier, M. N., Arnal, P. J., Wainrib, G., and Gramfort, A. (2018). **A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series.** *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):758–769.

[Courty et al., 2016] Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016). **Optimal transport for domain adaptation.** *Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

[Damodaran et al., 2018] Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018). **Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation.**

[Fernando et al., 2013] Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. (2013). **Unsupervised visual domain adaptation using subspace alignment.** In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967.

[Flamary et al., 2019] Flamary, R., Lounici, K., and Ferrari, A. (2019). **Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation.** *arXiv preprint arXiv:1905.10155*.

[Ganin et al., 2016] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). **Domain-adversarial training of neural networks.** *Journal of Machine Learning Research*, 17:1–35.

References ii

[Gnassounou et al., 2025] Gnassounou, T., Collas, A., Flamary, R., and Gramfort, A. (2025). **Psdnorm: Test-time temporal normalization for deep learning in sleep staging.** *arXiv preprint arXiv:2503.04582*.

[Gnassounou et al., 2024] Gnassounou, T., Collas, A., Flamary, R., Lounici, K., and Gramfort, A. (2024). **Multi-source and test-time domain adaptation on multivariate signals using spatio-temporal monge alignment.** *arXiv preprint arXiv:2407.14303*.

[Gnassounou et al., 2023] Gnassounou, T., Flamary, R., and Gramfort, A. (2023). **Convolutional monge mapping normalization for learning on biosignals.** In *Neural Information Processing Systems*.

[Gretton et al., 2009] Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). **Covariate shift by kernel mean matching.** *Dataset shift in machine learning*, 3(4):5.

[Hu et al., 2023] Hu, D., Liang, J., Liew, J. H., Xue, C., Bai, S., and Wang, X. (2023). **Mixed samples as probes for unsupervised model selection in domain adaptation.** In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 37923–37941. Curran Associates, Inc.

[Huang et al., 2006] Huang, J., Gretton, A., Borgwardt, K. M., Schölkopf, B., and Smola, A. J. (2006). **Correcting sample selection bias by unlabeled data.** In *Advances in neural information processing systems*, pages 601–608.

[Jin et al., 2020] Jin, Y., Wang, X., Long, M., and Wang, J. (2020). **Minimum class confusion for versatile domain adaptation.**

References iii

[Kang et al., 2019] Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. (2019). **Contrastive adaptation network for unsupervised domain adaptation.** In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4893–4902.

[Lalou et al., 2025] Lalou, Y., Gnassounou, T., Collas, A., de Mathelin, A., Kachaiev, O., Odonnat, A., Gramfort, A., Moreau, T., and Flamary, R. (2025). **Skada-bench: Benchmarking unsupervised domain adaptation methods with realistic validation.** *arXiv preprint arXiv:2407.11676*.

[Lipton et al., 2018] Lipton, Z., Wang, Y.-X., and Smola, A. (2018). **Detecting and correcting for label shift with black box predictors.** In *International conference on machine learning*, pages 3122–3130. PMLR.

[Long et al., 2015] Long, M., Cao, Y., Wang, J., and Jordan, M. I. (2015). **Learning transferable features with deep adaptation networks.**

[Montesuma and Mboula, 2021] Montesuma, E. F. and Mboula, F. M. N. (2021). **Wasserstein barycenter for multi-source domain adaptation.** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16785–16793.

[Moreau et al., 2022] Moreau, T., Massias, M., Gramfort, A., Ablin, P., Bannier, P.-A., Charlier, B., Dagréou, M., Dupre la Tour, T., Durif, G., Dantas, C. F., et al. (2022). **Benchopt: Reproducible, efficient and collaborative optimization benchmarks.** *Advances in Neural Information Processing Systems*, 35:25404–25421.

References iv

[Moreno-Torres et al., 2012] Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. (2012). **A unifying view on dataset shift in classification.** *Pattern recognition*, 45(1):521–530.

[Morerio et al., 2017] Morerio, P., Cavazza, J., and Murino, V. (2017). **Minimal-entropy correlation alignment for unsupervised deep domain adaptation.**

[Musgrave et al., 2021] Musgrave, K., Belongie, S., and Lim, S.-N. (2021). **Unsupervised domain adaptation: A reality check.** *arXiv preprint arXiv:2111.15672*.

[Pan et al., 2010] Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2010). **Domain adaptation via transfer component analysis.** *IEEE transactions on neural networks*, 22(2):199–210.

[Peng et al., 2019] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. (2019). **Moment matching for multi-source domain adaptation.** In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415.

[Perslev et al., 2021] Perslev, M., Darkner, S., Kempfner, L., Nikolic, M., Jennum, P. J., and Igel, C. (2021). **U-sleep: resilient high-frequency sleep staging.** *NPJ digital medicine*, 4(1):72.

[Redko et al., 2019] Redko, I., Courty, N., Flamary, R., and Tuia, D. (2019). **Optimal transport for multi-source domain adaptation under target shift.** In *International Conference on Artificial Intelligence and Statistics (AISTAT)*.

References v

[Saito et al., 2021] Saito, K., Kim, D., Teterwak, P., Sclaroff, S., Darrell, T., and Saenko, K. (2021). **Tune it the right way: Unsupervised validation of domain adaptation via soft neighborhood density.** In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9164–9173.

[Shimodaira, 2000] Shimodaira, H. (2000). **Improving predictive inference under covariate shift by weighting the log-likelihood function.** In *Journal of statistical planning and inference*, volume 90, pages 227–244. Elsevier.

[Si et al., 2010] Si, S., Tao, D., and Geng, B. (2010). **Bregman divergence-based regularization for transfer subspace learning.** *IEEE Transactions on Knowledge and Data Engineering*, 22(7):929–942.

[Sugiyama et al., 2005] Sugiyama, M., Krauledat, M., and Müller, K.-R. (2005). **Input-dependent estimation of generalization error under covariate shift.** In *Statistics and decisions*, volume 23, pages 249–279.

[Sugiyama et al., 2007] Sugiyama, M., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. (2007). **Covariate shift adaptation by importance weighted cross validation.** In *Journal of Machine Learning Research*, volume 8, pages 985–1005.

[Sugiyama et al., 2012] Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). **Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation.** *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044.

[Sun and Saenko, 2016] Sun, B. and Saenko, K. (2016). **Deep CORAL: Correlation Alignment for Deep Domain Adaptation, pages 443–450.** Springer International Publishing, Cham.

References vi

[Tzeng et al., 2014] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). **Deep domain confusion: Maximizing for domain invariance.** *arXiv preprint arXiv:1412.3474*.

[Xiao et al., 2023] Xiao, Z., Wang, H., Jin, Y., Feng, L., Chen, G., Huang, F., and Zhao, J. (2023). **Spa: A graph spectral alignment perspective for domain adaptation.**

[Xie et al., 2024] Xie, R., Odonnat, A., Feofanov, V., Deng, W., Zhang, J., and An, B. (2024). **Mano: Exploiting matrix norm for unsupervised accuracy estimation under distribution shifts.** *arXiv preprint arXiv:2405.18979*.

[You et al., 2019] You, K., Wang, X., Long, M., and Jordan, M. (2019). **Towards accurate model selection in deep unsupervised domain adaptation.** In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7124–7133. PMLR.

[Zhang et al., 2013] Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. (2013). **Domain adaptation under target and conditional shift.** In *International Conference on Machine Learning*, pages 819–827. PMLR.

[Zhang et al., 2019] Zhang, Y., Liu, T., Long, M., and Jordan, M. I. (2019). **Bridging theory and algorithm for domain adaptation.**

[Zhou et al., 2022] Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. (2022). **Domain generalization: A survey.** *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4396–4415.

[Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). **Unpaired image-to-image translation using cycle-consistent adversarial networks.** In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.